

Oracle® Solaris 管理：安全服务

版权所有 © 2002, 2012, Oracle 和/或其附属公司。保留所有权利。

本软件和相关文档是根据许可证协议提供的，该许可证协议中规定了关于使用和公开本软件和相关文档的各种限制，并受知识产权法的保护。除非在许可证协议中明确许可或适用法律明确授权，否则不得以任何形式、任何方式使用、拷贝、复制、翻译、广播、修改、授权、传播、分发、展示、执行、发布或显示本软件和相关文档的任何部分。除非法律要求实现互操作，否则严禁对本软件进行逆向工程设计、反汇编或反编译。

此文档所含信息可能随时被修改，恕不另行通知，我们不保证该信息没有错误。如果贵方发现任何问题，请书面通知我们。

如果将本软件或相关文档交付给美国政府，或者交付给以美国政府名义获得许可证的任何机构，必须符合以下规定：

U.S. GOVERNMENT END USERS:

Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

本软件或硬件是为了在各种信息管理应用领域内的一般使用而开发的。它不应被应用于任何存在危险或潜在危险的应用领域，也不是为此而开发的，其中包括可能会产生人身伤害的应用领域。如果在危险应用领域内使用本软件或硬件，贵方应负责采取所有适当的防范措施，包括备份、冗余和其它确保安全使用本软件或硬件的措施。对于因在危险应用领域内使用本软件或硬件所造成的一切损失或损害，Oracle Corporation 及其附属公司概不负责。

Oracle 和 Java 是 Oracle 和/或其附属公司的注册商标。其他名称可能是各自所有者的商标。

Intel 和 Intel Xeon 是 Intel Corporation 的商标或注册商标。所有 SPARC 商标均是 SPARC International, Inc 的商标或注册商标，并应按照许可证的规定使用。AMD、Opteron、AMD 徽标以及 AMD Opteron 徽标是 Advanced Micro Devices 的商标或注册商标。UNIX 是 The Open Group 的注册商标。

本软件或硬件以及文档可能提供了访问第三方内容、产品和服务的方式或有关这些内容、产品和服务的信息。对于第三方内容、产品和服务，Oracle Corporation 及其附属公司明确表示不承担任何种类的担保，亦不对其承担任何责任。对于因访问或使用第三方内容、产品和服务所造成的任何损失、成本或损害，Oracle Corporation 及其附属公司概不负责。

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
加密服务	30
验证服务	31
使用加密的验证	31
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
控制对计算机系统的访问	35
维护物理安全性	36
维护登录控制	36
控制对设备的访问	40
设备策略（概述）	41
设备分配（概述）	42
控制对计算机资源的访问	42
限制和监视超级用户	42
配置基于角色的访问控制以替换超级用户	42
防止无意中误用系统资源	43
限制 setuid 可执行文件	44
使用缺省安全 (Secure by Default) 配置	44

使用资源管理功能	44
使用 Oracle Solaris 区域	45
监视计算机资源的使用情况	45
监视文件完整性	45
控制对文件的访问	45
通过加密保护文件	46
使用访问控制列表	46
跨计算机共享文件	46
限制对共享文件的 root 访问	47
控制网络访问	47
网络安全性机制	47
远程访问的验证和授权	48
防火墙系统	49
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）	54
保证登录和口令的安全（任务列表）	54
▼ 如何更改 root 口令	54
▼ 如何显示用户的登录状态	55
▼ 如何显示没有口令的用户	56
▼ 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	57
更改口令加密的缺省算法（任务）	59
▼ 如何指定口令加密的算法	59
▼ 如何为 NIS 域指定新的口令算法	60
▼ 如何为 LDAP 域指定新的口令算法	60
监视和限制超级用户（任务）	61
▼ 如何监视正在使用 su 命令的用户	61
▼ 如何限制和监视超级用户登录	62
控制对系统硬件的访问（任务）	63
▼ 如何要求提供用于访问硬件的口令	64

▼ 如何禁用系统的异常中止序列	64
4 病毒扫描服务（任务）	67
关于病毒扫描	67
关于 Vscan 服务	68
使用 Vscan 服务（任务）	68
▼ 如何在文件系统上启用病毒扫描	69
▼ 如何启用 Vscan 服务	69
▼ 如何添加扫描引擎	70
▼ 如何查看 Vscan 属性	70
▼ 如何更改 Vscan 属性	70
▼ 如何从病毒扫描操作中排除文件	71
5 控制对设备的访问（任务）	73
配置设备（任务列表）	73
配置设备策略（任务）	74
配置设备策略（任务列表）	74
▼ 如何查看设备策略	74
▼ 如何更改现有设备上的设备策略	75
▼ 如何审计设备策略的更改	76
▼ 如何从 /dev/* 设备检索 IP MIB-II 信息	76
管理设备分配（任务）	76
管理设备分配（任务列表）	77
▼ 如何启用设备分配	77
▼ 如何授权用户分配设备	78
▼ 如何查看有关设备的分配信息	78
▼ 强制分配设备	79
▼ 强制解除分配设备	79
▼ 如何更改可分配的设备	80
▼ 如何审计设备分配	81
分配设备（任务）	81
▼ 如何分配设备	81
▼ 如何挂载分配的设备	82
▼ 如何解除分配设备	84
设备保护（参考）	84

设备策略命令	85
设备分配	85
6 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）	96
BART 安全注意事项	96
使用 BART（任务列表）	96
▼ 如何创建清单	97
▼ 如何定制清单	98
▼ 如何比较同一个系统在一段时间内的清单	99
▼ 如何比较不同系统的清单	101
▼ 如何通过指定文件属性定制 BART 报告	103
▼ 如何通过使用 Rules 文件定制 BART 报告	104
BART 清单、Rules 文件和报告（参考）	105
BART 清单文件格式	105
BART Rules 文件格式	106
BART 报告	107
7 控制对文件的访问（任务）	109
使用 UNIX 权限保护文件	109
用于查看和保证文件安全的命令	109
文件和目录的所有权	110
UNIX 文件权限	110
特殊文件权限（setuid、setgid 和 Sticky 位）	111
缺省 umask 值	112
文件权限模式	113
使用访问控制列表保护 UFS 文件	115
防止可执行文件危及安全	116
保护文件（任务）	116
使用 UNIX 权限保护文件（任务列表）	117
▼ 如何显示文件信息	117
▼ 如何更改文件的所有者	118

▼ 如何更改文件的组所有权	119
▼ 如何在符号模式下更改文件权限	119
▼ 如何在绝对模式下更改文件权限	120
▼ 如何在绝对模式下更改特殊文件权限	121
防止程序受到安全风险（任务列表）	122
▼ 如何查找具有特殊文件权限的文件	123
▼ 如何禁止程序使用可执行栈	124
 第 3 部分 角色、权限配置文件和特权	125
 8 使用角色和特权（概述）	127
基于角色的访问控制（概述）	127
RBAC：超级用户模型的替代方法	127
RBAC 元素和基本概念	130
特权升级	132
RBAC 授权	133
授权和特权	134
特权应用程序和 RBAC	134
RBAC 权限配置文件	135
RBAC 角色	135
配置文件 Shell 和 RBAC	136
名称服务范围和 RBAC	137
直接指定安全属性时的安全注意事项	137
直接指定安全属性时的可用性注意事项	137
特权（概述）	138
特权保护内核进程	138
特权说明	139
具有特权与不具有特权的系统之间的管理差别	139
特权和系统资源	140
如何实现特权	141
进程如何获取特权	142
指定特权	142
特权和设备	144
特权和调试	144

9 使用基于角色的访问控制（任务）	145
使用 RBAC（任务）	145
查看和使用 RBAC 缺省值（任务）	146
查看和使用 RBAC 缺省值（任务列表）	146
▼如何查看所有已定义的安全属性	146
▼如何查看指定的权限	147
▼如何承担角色	149
▼如何获取管理权限	150
为站点定制 RBAC（任务）	153
初次配置 RBAC（任务列表）	153
▼如何规划 RBAC 实现	153
▼如何创建角色	155
▼如何指定角色	157
▼如何审计角色	159
▼如何创建或更改权限配置文件	159
▼如何为传统应用程序添加 RBAC 属性	161
▼如何排除 RBAC 和特权指定故障	162
管理 RBAC（任务）	165
管理 RBAC（任务列表）	165
▼如何更改角色的口令	166
▼如何更改角色的安全属性	167
▼如何更改用户的 RBAC 属性	168
▼如何将用户限于桌面应用程序	170
▼如何将管理员限于显式指定的权限	171
▼如何让用户使用自己的口令承担角色	172
▼如何将 root 角色更改为用户	173
使用特权（任务）	175
确定您的特权（任务列表）	175
▼如何列出系统上的特权	176
▼如何确定已直接指定给您的特权	176
▼如何确定您可以运行的特权命令	178
管理特权（任务列表）	179
▼如何确定进程的特权	180
▼如何确定程序所需的特权	181
▼如何运行具有特权命令的 Shell 脚本	183

10 Oracle Solaris 中的安全属性（参考）	185
权限配置文件	185
查看权限配置文件的内容	187
搜索指定安全属性的顺序	187
Authorizations（授权）	188
授权命名约定	188
授权粒度示例	188
授权中的委托授权	188
RBAC 数据库	189
RBAC 数据库和命名服务	189
user_attr 数据库	189
auth_attr 数据库	190
prof_attr 数据库	190
exec_attr 数据库	191
policy.conf 文件	191
RBAC 命令	192
管理 RBAC 的命令	192
要求授权的选定命令	192
特权	194
用于处理特权的管理命令	194
包含特权信息的文件	194
特权和审计	195
防止特权升级	195
传统应用程序和特权模型	196
 第 4 部分 加密服务	 197
 11 加密框架（概述）	 199
加密框架简介	199
加密框架术语	201
加密框架的范围	202
加密框架中的管理命令	203
加密框架中的用户级命令	203
第三方软件的二进制文件签名	203
加密框架的插件	204

加密服务和区域	204
12 加密框架（任务）	205
使用加密框架（任务列表）	205
使用加密框架保护文件（任务）	205
使用加密框架保护文件（任务列表）	205
▼ 如何使用 <code>dd</code> 命令生成对称密钥	206
▼ 如何使用 <code>pktool</code> 命令生成对称密钥	208
▼ 如何计算文件摘要	212
▼ 如何计算文件的 MAC	213
▼ 如何加密和解密文件	215
管理加密框架（任务）	218
管理加密框架（任务列表）	218
▼ 如何列出可用提供者	218
▼ 如何添加软件提供者	222
▼ 如何禁止使用用户级机制	223
▼ 如何禁止使用内核软件提供者	225
▼ 如何列出硬件提供者	227
▼ 如何禁用硬件提供者机制和功能	228
▼ 如何刷新或重新启动所有加密服务	230
13 密钥管理框架	231
管理公钥技术	231
密钥管理框架实用程序	232
KMF 策略管理	232
KMF 插件管理	232
KMF 密钥库管理	233
使用密钥管理框架（任务）	233
使用密钥管理框架（任务列表）	233
▼ 如何使用 <code>pktool gencert</code> 命令创建证书	234
▼ 如何将证书导入密钥库	235
▼ 如何以 PKCS #12 格式导出证书和私钥	237
▼ 如何使用 <code>pktool setpin</code> 命令生成口令短语	238
▼ 如何使用 <code>pktool genkeypair</code> 命令生成密钥对	238
▼ 如何使用 <code>pktool signcsr</code> 命令签署证书请求	242

▼ 如何在 KMF 中管理第三方插件	243
第 5 部分 验证服务和安全通信	245
14 网络服务验证（任务）	247
安全 RPC 概述	247
NFS 服务和安全 RPC	247
使用安全 NFS 的 DES 加密	248
Kerberos 验证	248
Diffie-Hellman 验证和安全 RPC	248
使用安全 RPC 管理验证（任务）	251
管理安全 RPC（任务列表）	251
▼ 如何重新启动安全 RPC Keyserver	251
▼ 如何为 NIS 主机设置 Diffie-Hellman 密钥	252
▼ 如何为 NIS 用户设置 Diffie-Hellman 密钥	253
▼ 如何通过 Diffie-Hellman 验证共享 NFS 文件	254
15 使用 PAM	255
PAM（概述）	255
使用 PAM 的益处	255
PAM 框架介绍	256
此发行版中 PAM 的更改	257
PAM（任务）	257
PAM（任务列表）	257
规划 PAM 实现	257
▼ 如何添加 PAM 模块	258
▼ 如何使用 PAM 防止从远程系统进行 Rhost 样式的访问	259
▼ 如何记录 PAM 错误报告	259
PAM 配置（参考）	259
PAM 配置文件语法	259
PAM 堆栈工作原理	260
PAM 堆栈示例	263

16 使用 SASL	265
SASL (概述)	265
SASL (参考)	265
SASL 插件	266
SASL 环境变量	266
SASL 选项	266
 17 使用 安全 Shell (任务)	 269
安全 Shell (概述)	269
安全 Shell 验证	270
企业中的安全 Shell	271
安全 Shell 和 OpenSSH 项目	271
安全 Shell 和 FIPS-140 支持	272
安全 Shell (任务列表)	272
配置安全 Shell (任务)	273
配置 安全 Shell (任务列表)	273
▼ 如何为安全 Shell 设置基于主机的验证	273
▼ 如何在安全 Shell 中配置端口转发	275
▼ 如何创建不同于 SSH 系统缺省值的用户和主机例外	276
使用 安全 Shell (任务)	277
使用 安全 Shell (任务列表)	277
▼ 如何生成用于安全 Shell 的公钥/私钥对	277
▼ 如何更改安全 Shell 私钥的口令短语	279
▼ 如何使用安全 Shell 登录到远程主机	279
▼ 如何减少安全 Shell 中的口令提示	280
▼ 如何在安全 Shell 中使用端口转发	282
▼ 如何使用安全 Shell 复制文件	283
▼ 如何设置到防火墙外部主机的缺省连接	283
 18 安全 Shell (参考)	 287
典型的安全 Shell 会话	287
安全 Shell 中的会话特征	287
安全 Shell 中的验证和密钥交换	288
安全 Shell 中的命令执行和数据转发	289
安全 Shell 中的客户机和服务器配置	289

安全 Shell 中的客户机配置	289
安全 Shell 中的服务器配置	289
安全 Shell 中的关键字	290
安全 Shell 中主机特定的参数	294
安全 Shell 和登录环境变量	294
在安全 Shell 中维护已知主机	295
安全 Shell 文件	295
安全 Shell 命令	297
 第 6 部分 Kerberos 服务	299
 19 Kerberos 服务介绍	301
什么是 Kerberos 服务?	301
Kerberos 服务的工作原理	302
初始验证：票证授予票证	303
后续 Kerberos 验证	304
Kerberos 远程应用程序	305
Kerberos 主体	305
Kerberos 领域	306
Kerberos 服务器	307
Kerberos 安全服务	308
各种 Kerberos 发行版的组件	308
Kerberos 组件	309
关于 Oracle Solaris 11 发行版中的 Kerberos	310
 20 规划 Kerberos 服务	313
为什么要规划 Kerberos 部署?	313
规划 Kerberos 领域	314
领域名称	314
领域数量	314
领域分层结构	314
将主机名映射到领域	315
客户机与服务主体名称	315
KDC 端口和管理服务端口	316

从 KDC 数量	316
将 GSS 凭证映射到 UNIX 凭证	317
自动将用户迁移到 Kerberos 领域	317
要使用的数据库传播系统	317
领域内的时钟同步	318
客户机配置选项	318
提高客户机登录安全性	319
KDC 配置选项	319
委托服务的信任	319
Kerberos 加密类型	320
图形化 Kerberos 管理工具中的联机帮助 URL	320
21 配置 Kerberos 服务（任务）	321
配置 Kerberos 服务（任务列表）	321
配置额外 Kerberos 服务（任务列表）	322
配置 KDC 服务器	322
▼ 如何自动配置主 KDC	323
▼ 如何交互配置主 KDC	324
▼ 如何手动配置主 KDC 服务器	325
▼ 如何配置 KDC 以使用 LDAP 数据服务器	329
▼ 如何自动配置从 KDC	335
▼ 如何交互配置从 KDC	336
▼ 如何手动配置从 KDC 服务器	337
▼ 如何刷新主服务器上的票证授予服务密钥	340
配置跨领域验证	341
▼ 如何建立层次化跨领域验证	341
▼ 如何建立直接跨领域验证	342
配置 Kerberos 网络应用服务器	343
▼ 如何配置 Kerberos 网络应用服务器	343
▼ 运行 FTP 时如何使用带有 Kerberos 的通用安全服务	345
配置 Kerberos NFS 服务器	346
▼ 如何配置 Kerberos NFS 服务器	346
▼ 如何创建凭证表	348
▼ 如何向凭证表中添加单个项	348
▼ 如何提供各领域之间的凭证映射	349

▼ 如何使用多种 Kerberos 安全模式设置安全的 NFS 环境	350
配置 Kerberos 客户机	351
配置 Kerberos 客户机（任务列表）	351
▼ 如何创建 Kerberos 客户机安装配置文件	352
▼ 如何自动配置 Kerberos 客户机	353
▼ 如何交互配置 Kerberos 客户机	354
▼ 如何为 Active Directory 服务器配置 Kerberos 客户机	357
▼ 如何手动配置 Kerberos 客户机	358
▼ 如何禁用票务授予票证的验证	362
▼ 如何以 root 用户身份访问受 Kerberos 保护的 NFS 文件系统	363
▼ 如何在 Kerberos 领域中配置用户自动迁移	364
▼ 如何配置帐户锁定	366
同步 KDC 与 Kerberos 客户机的时钟	367
交换主 KDC 服务器与从 KDC 服务器	368
▼ 如何配置可交换的从 KDC 服务器	368
▼ 如何交换主 KDC 服务器与从 KDC 服务器	369
管理 Kerberos 数据库	372
备份和传播 Kerberos 数据库	372
▼ 如何备份 Kerberos 数据库	374
▼ 如何恢复 Kerberos 数据库	374
▼ 如何在服务升级后转换 Kerberos 数据库	375
▼ 如何重新配置主 KDC 服务器以使用增量传播	376
▼ 如何重新配置从 KDC 服务器以使用增量传播	377
▼ 如何配置从 KDC 服务器以使用完全传播	378
▼ 如何验证 KDC 服务器是否已同步	381
▼ 如何手动将 Kerberos 数据库传播到从 KDC 服务器	382
设置并行传播	383
设置并行传播的配置步骤	384
管理存储文件	385
▼ 如何删除存储文件	385
▼ 如何使用新的主密钥	385
在 LDAP 目录服务器上管理 KDC	387
▼ 如何在非 Kerberos 对象类类型中混合 Kerberos 主体属性	387
▼ 如何在 LDAP 目录服务器上销毁领域	388
增强 Kerberos 服务器的安全性	389
▼ 如何仅启用基于 Kerberos 的应用程序	389

▼ 如何限制对 KDC 服务器的访问	389
▼ 如何使用字典文件提高口令的安全性	390
22 Kerberos 错误消息和故障排除	393
Kerberos 错误消息	393
SEAM Tool 错误消息	393
常见的 Kerberos 错误消息 (A-M)	394
常见的 Kerberos 错误消息 (N-Z)	403
Kerberos 故障排除	406
▼ 如何识别密钥版本号相关的问题	406
krb5.conf 文件的格式存在问题	407
传播 Kerberos 数据库时出现问题	407
挂载基于 Kerberos 的 NFS 文件系统时出现问题	407
以 root 用户身份进行验证时出现问题	408
观察从 GSS 凭证到 UNIX 凭证的映射	408
对 Kerberos 服务使用 DTrace	409
23 管理 Kerberos 主体和策略 (任务)	411
管理 Kerberos 主体和策略的方法	411
SEAM Tool	412
SEAM Tool 的等效命令行	412
SEAM Tool 修改的唯一文件	413
SEAM Tool 的打印和联机帮助功能	413
在 SEAM Tool 中处理大型列表	414
▼ 如何启动 SEAM Tool	415
管理 Kerberos 主体	416
管理 Kerberos 主体 (任务列表)	416
自动创建新的 Kerberos 主体	416
▼ 如何查看 Kerberos 主体列表	417
▼ 如何查看 Kerberos 主体属性	419
▼ 如何创建新的 Kerberos 主体	421
▼ 如何复制 Kerberos 主体	424
▼ 如何修改 Kerberos 主体	424
▼ 如何删除 Kerberos 主体	425
▼ 如何设置缺省值以创建新的 Kerberos 主体	426

▼ 如何修改 Kerberos 管理特权	427
管理 Kerberos 策略	429
管理 Kerberos 策略（任务列表）	429
▼ 如何查看 Kerberos 策略列表	429
▼ 如何查看 Kerberos 策略属性	431
▼ 如何创建新的 Kerberos 策略	433
▼ 如何复制 Kerberos 策略	435
▼ 如何修改 Kerberos 策略	435
▼ 如何删除 Kerberos 策略	436
SEAM Tool 参考	437
SEAM Tool 面板说明	437
以受限 Kerberos 管理特权使用 SEAM Tool	440
管理密钥表文件	441
管理密钥表文件（任务列表）	442
▼ 如何将 Kerberos 服务主体添加至密钥表文件	442
▼ 如何从密钥表文件中删除服务主体	443
▼ 如何显示密钥表文件中的密钥列表（主体）	444
▼ 如何在主机上临时禁用对服务的验证	445
24 使用 Kerberos 应用程序（任务）	447
Kerberos 票证管理	447
是否需要担心票证？	447
创建 Kerberos 票证	448
查看 Kerberos 票证	448
销毁 Kerberos 票证	450
Kerberos 口令管理	450
关于选择口令的建议	450
更改口令	451
授予对帐户的访问权限	453
Kerberos 用户命令	455
基于 Kerberos 的命令概述	455
转发 Kerberos 票证	457
使用基于 Kerberos 的命令（示例）	459

25 Kerberos 服务 (参考)	461
Kerberos 文件	461
Kerberos 命令	462
Kerberos 守护进程	463
Kerberos 术语	464
特定于 Kerberos 的术语	464
特定于验证的术语	464
票证类型	465
Kerberos 验证系统的工作原理	468
Kerberos 服务如何与 DNS 和 nsswitch 服务交互	469
使用 Kerberos 获取服务访问权限	469
获取用于票证授予服务的凭证	469
获取服务器凭证	470
获取对特定服务的访问权限	471
使用 Kerberos 加密类型	472
使用 gsscred 表	473
Oracle Solaris Kerberos 和 MIT Kerberos 之间的显著差异	474
 第 7 部分 在 Oracle Solaris 中审计	 475
 26 审计 (概述)	 477
什么是审计?	477
审计术语和概念	478
审计事件	480
审计类和预选	481
审计记录和审计标记	481
审计插件模块	482
审计日志	482
存储和管理审计迹	484
确保时间戳可靠	484
管理远程系统信息库	485
审计如何与安全相关?	485
审计工作原理	485
配置审计的方式	486
在具有 Oracle Solaris 区域的系统上审计	487

关于此发行版中的审计服务	488
27 规划审计	491
规划审计（任务）	491
▼ 如何规划区域中的审计	492
▼ 如何规划审计记录的存储	493
▼ 如何规划要审计的对象及内容	494
了解审计策略	496
控制审计成本	498
延长审计数据处理时间产生的成本	498
分析审计数据产生的成本	498
存储审计数据产生的成本	499
有效审计	499
28 管理审计（任务）	501
管理审计（任务列表）	501
配置审计服务（任务）	502
配置审计服务（任务列表）	502
▼ 如何显示审计服务缺省值	503
▼ 如何预选审计类	504
▼ 如何配置用户审计特征	505
▼ 如何更改审计策略	508
▼ 如何更改审计队列控制	511
▼ 如何配置 <code>audit_warn</code> 电子邮件别名	512
▼ 如何添加审计类	513
▼ 如何更改审计事件的类成员身份	514
配置审计日志（任务）	515
配置审计日志（任务列表）	515
▼ 如何为审计文件创建 ZFS 文件系统	515
▼ 如何为审计迹指定审计空间	518
▼ 如何向远程系统信息库发送审计文件	520
▼ 如何配置 <code>syslog</code> 审计日志	521
在区域中配置审计服务（任务）	523
▼ 如何配置以相同方式审计所有区域	523
▼ 如何配置每区域审计	525

启用和禁用审计服务（任务）	526
▼ 如何刷新审计服务	526
▼ 如何禁用审计服务	528
▼ 如何启用审计服务	529
在本地系统上管理审计记录（任务）	530
在本地系统上管理审计记录（任务列表）	530
▼ 如何显示审计记录定义	530
▼ 如何合并审计迹中的审计文件	532
▼ 如何从审计迹中选择审计事件	533
▼ 如何查看二进制审计文件的内容	535
▼ 如何清除 <code>not_terminated</code> 审计文件	537
▼ 如何防止审计迹溢出	538
审计服务的故障排除（任务）	539
审计服务的故障排除（任务列表）	539
▼ 如何确定审计正在运行	540
▼ 如何减少生成的审计记录数目	542
▼ 如何审计用户执行的所有命令	544
▼ 如何找到对特定文件更改的审计记录	545
▼ 如何更新已登录用户的预选掩码	547
▼ 如何阻止审计特定事件	548
▼ 如何限制二进制审计文件的大小	549
▼ 如何压缩专用文件系统上的审计文件	549
▼ 如何审计从其他操作系统的登录	550
▼ 如何审计 FTP 和 SFTP 文件传输	551
 29 审计（参考）	553
审计服务	553
审计服务手册页	554
用于管理审计的权限配置文件	555
审计和 Oracle Solaris 区域	556
审计类	556
审计类语法	557
审计插件	557
审计策略	558
同步事件和异步事件的审计策略	558

进程审计特征	559
审计迹	560
二进制审计文件名称约定	560
审计记录结构	560
审计记录分析	560
审计标记格式	561
acl 标记	563
argument 标记	563
attribute 标记	563
cmd 标记	563
exec_args 标记	564
exec_env 标记	564
file 标记	564
fmri 标记	565
group 标记	565
header 标记	565
ip address 标记	566
ip port 标记	566
ipc 标记	566
IPC_perm 标记	567
path 标记	567
path_attr 标记	567
privilege 标记	567
process 标记	568
return 标记	568
sequence 标记	568
socket 标记	568
subject 标记	569
text 标记	569
trailer 标记	569
use of authorization 标记	570
use of privilege 标记	570
user 标记	570
xclient 标记	570
zonename 标记	570

词汇表 571

索引 581

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 Oracle Solaris 操作系统 (Oracle Solaris OS) 管理信息的重要部分。本书假定您已经安装了最新发行版，并且设置了计划使用的任何网络软件。Oracle Solaris OS 属于 Oracle Solaris 产品系列，该系列包含安全 Shell 等许多功能。

注 – 此 Oracle Solaris 发行版支持使用 SPARC 和 x86 系列处理器体系结构的系统。支持的系统可以在 [Oracle Solaris OS: Hardware Compatibility Lists](#)（Oracle Solaris OS：硬件兼容性列表）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。

目标读者

本书适用于所有负责管理一个或多个运行 Oracle Solaris 的系统的人员。要使用本书，您应当有两年以上 UNIX 系统管理经验。参加 UNIX 系统管理培训课程可能会对您有所帮助。

系统管理指南系列书籍的结构

下表列出了系统管理指南系列中各本书包含的主题。

书名	主题
《在 SPARC 平台上引导和关闭 Oracle Solaris》	引导和关闭系统、管理引导服务、修改引导行为、从 ZFS 引导、管理引导归档文件以及在 SPARC 平台上进行引导故障排除
《在 x86 平台上引导和关闭 Oracle Solaris》	引导和关闭系统、管理引导服务、修改引导行为、从 ZFS 引导、管理引导归档文件以及在 x86 平台上进行引导故障排除
《Oracle Solaris 管理：常见任务》	使用 Oracle Solaris 命令；引导和关闭系统；管理用户帐户和组；管理服务、硬件故障、系统信息、系统资源和系统性能；管理软件、打印、控制台和终端；以及排除系统和软件的故障
《Oracle Solaris 管理：设备和文件系统》	可移除介质、磁盘和设备、文件系统以及备份和还原数据

书名	主题
《Oracle Solaris 管理：IP 服务》	TCP/IP 网络管理、IPv4 和 IPv6 地址管理、DHCP、IPsec、IKE、IP 过滤器和 IPQoS
《Oracle Solaris Administration: Naming and Directory Services》	DNS、NIS 和 LDAP 命名与目录服务，包括从 NIS 转换至 LDAP
《Oracle Solaris 管理：网络接口和网络虚拟化》	自动和手动 IP 接口配置包括 WiFi 无线、网桥管理、VLAN、聚合、LLDP 和 IPMP、虚拟 NIC 和资源管理。
《Oracle Solaris 管理：网络服务》	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
《Oracle Solaris 管理：Oracle Solaris Zones、Oracle Solaris 10 Zones 和资源管理》	资源管理功能，可用于控制应用程序对可用系统资源的使用；Oracle Solaris Zones 软件分区技术，用于虚拟化操作系统服务以便为正在运行的应用程序创建隔离环境；Oracle Solaris 10 Zones，用于托管运行在 Oracle Solaris 11 内核上的 Oracle Solaris 10 环境
《Oracle Solaris 管理：安全服务》	审计，设备管理，文件安全，BART，Kerberos 服务，PAM，加密框架，密钥管理，特权，RBAC，SASL，安全 Shell 和病毒扫描
《Oracle Solaris Administration: SMB and Windows Interoperability》	SMB 服务，可用于配置 Oracle Solaris 系统，以使 SMB 客户机可使用 SMB 共享；SMB 客户机，可用于访问 SMB 共享；本机标识映射服务，可用于在 Oracle Solaris 系统和 Windows 系统之间映射用户和组的标识
《Oracle Solaris 管理：ZFS 文件系统》	ZFS 存储池和文件系统的创建和管理、快照、克隆、备份；使用访问控制列表 (access control list, ACL) 保护 ZFS 文件；在安装了区域的 Oracle Solaris 系统上使用 ZFS
《Trusted Extensions 配置和管理》	专用于 Trusted Extensions 的系统安装、配置和管理
《Oracle Solaris 11 安全准则》	保护 Oracle Solaris 系统，及其安全性功能（例如区域、ZFS 和 Trusted Extensions）的使用方案
《从 Oracle Solaris 10 转换到 Oracle Solaris 11》	在以下领域为从 Oracle Solaris 10 转换至 Oracle Solaris 11 提供系统管理信息和示例：安装、设备、磁盘和文件系统管理、软件管理、联网、系统管理、安全性、虚拟化、桌面功能、用户帐户管理、用户环境仿真卷、以及故障排除和数据恢复

获取 Oracle 支持

Oracle 客户可以通过 My Oracle Support 获取电子支持。有关信息，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>，或访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>（如果您听力受损）。

印刷约定

下表介绍了本书中的印刷约定。

表 P-1 印刷约定

字体或符号	含义	示例
AaBbCc123	命令、文件和目录的名称；计算机屏幕输出	编辑 <code>.login</code> 文件。 使用 <code>ls -a</code> 列出所有文件。 <code>machine_name% you have mail.</code>
AaBbCc123	用户键入的内容，与计算机屏幕输出的显示不同	<code>machine_name% su</code> <code>Password:</code>
<i>aabbcc123</i>	要使用实名或值替换的命令行占位符	删除文件的命令为 <code>rm filename</code> 。
<i>AaBbCc123</i>	保留未译的新词或术语以及要强调的词	这些称为 <i>Class</i> 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

下表显示了 Oracle Solaris OS 中包含的缺省 UNIX shell 系统提示符和超级用户提示符。请注意，在命令示例中显示的缺省系统提示符可能会有所不同，具体取决于 Oracle Solaris 发行版。

表 P-2 shell 提示符

shell	提示符
Bash shell、Korn shell 和 Bourne shell	\$
Bash shell、Korn shell 和 Bourne shell 超级用户	#
C shell	<code>machine_name%</code>
C shell 超级用户	<code>machine_name#</code>

第 1 部分

安全性概述

本书重点介绍可增强 Oracle Solaris OS 安全性的功能。本书适用于系统管理员和使用这些安全功能的用户。第 1 章，[安全性服务（概述）](#)一章介绍了本书中的主题。

安全性服务（概述）

为了维护 Oracle Solaris OS 的安全，本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
- 第 30 页中的“[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，以及管理加密提供者和公钥对象
- 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在通信时不会受到窃听、修改或电子欺骗
- 第 32 页中的“[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](#)”—为系统或系统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 Oracle Solaris 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 的访问以及限制哪些用户可以引导系统的命令。请参见第 63 页中的“[控制对系统硬件的访问（任务）](#)”。
- [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 42 页中的“[控制对计算机资源的访问](#)”。

有关管理 Oracle Solaris Zones 中资源的信息，请参见《[Oracle Solaris 管理：Oracle Solaris Zones、Oracle Solaris 10 Zones 和资源管理](#)》中的第 I 部分，“Oracle Solaris 资源管理”。

- **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 127 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限。这些进程权限在内核中执行。请参见第 138 页中的“[特权（概述）](#)”。
 - **设备管理**—为已受 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 40 页中的“[控制对设备的访问](#)”。
 - **基本审计报告工具 (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 6 章，[使用基本审计报告工具（任务）](#)。
 - **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 7 章，[控制对文件的访问（任务）](#)。
 - **防病毒软件**—vscan 服务在应用程序使用文件之前检查文件中是否存在病毒。在文件系统的任何客户机访问文件之前，文件系统可调用此服务以根据最新的病毒定义对文件进行即时扫描。
- 第三方应用程序执行此即时扫描。可在打开文件时和关闭文件后对其进行扫描。请参见第 4 章，[病毒扫描服务（任务）](#)。

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了验证和加密机制。

- **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，该框架基于 RSA Security Inc. 推出的 PKCS #11 加密令牌接口 (Cryptographic Token Interface, Cryptoki) 标准。其应用场合包括口令、IPsec 和第三方应用程序。该框架集中了用于加密的硬件和软件资源。PKCS #11 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 11 章，[加密框架（概述）](#)。
- **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 247 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 19 章，[Kerberos 服务介绍](#)。
 - 有关如何在安全 Shell 中使用 RSA、DSA 和密码（如 Blowfish）的信息，请参见第 17 章，[使用安全 Shell（任务）](#)。
 - 有关在口令中使用加密算法的信息，请参见第 59 页中的[“更改口令加密的缺省算法（任务）”](#)。

- 密钥管理框架 (Key Management Framework, KMF) 提供了一个用于管理公钥对象（包括策略、密钥和证书）的中心实用程序。KMF 基于 OpenSSL、NSS 和 PKCS #11 公钥技术来管理这些对象。请参见第 13 章，[密钥管理框架](#)。

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。例如，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。Oracle 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 挂载和命名服务（如 NIS）。请参见第 247 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 等。请参见第 15 章，[使用 PAM](#)。
- **简单验证和安全层 (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 16 章，[使用 SASL](#)。
- **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 17 章，[使用 安全 Shell（任务）](#)。
- **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 6 部分。

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。Oracle Solaris 用于安全通信的功能有：

- **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”（中间人）攻击。通过公钥密码学提供了强验证。X 窗口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 17 章，[使用 安全 Shell（任务）](#)。
- **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 6 部分。
- **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《[Oracle Solaris 管理：IP 服务](#)》中的第 III 部分，“[IP 安全性](#)”。

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 7 部分。

安全策略

本书中使用的短语“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合和相关措施，可用于定义所处理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，Oracle Solaris 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 Oracle Solaris 中“策略”一词的使用

“策略”术语	选择的手册页	详细信息
audit policy（审计策略）	auditconfig(1M)	第 26 章，审计（概述）
policy in the Cryptographic Framework	cryptoadm(1M)	第 11 章，加密框架（概述）
device policy	getdevpolicy(1M)	第 40 页中的“控制对设备的访问”
Kerberos policy（Kerberos 策略）	krb5.conf(4)	第 23 章，管理 Kerberos 主体和策略（任务）
network policies	ipfilter(5)、ipadm(1M)、ike.config(4)、ipsec.conf(4)、ipsecd(1M)	《Oracle Solaris 管理：故障排除》中的第 III 部分，“IP 安全性”
password policy	passwd(1)、crypt.conf(4)、policy.conf(4)	第 34 页中的“维护登录控制”
policy for public key technologies（公钥技术的策略）	kmfcfg(1)	第 13 章，密钥管理框架
RBAC policy	rbac(5)、policy.conf(4)	第 191 页中的“policy.conf 文件”

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）
- 第 3 章，控制对系统的访问（任务）
- 第 4 章，病毒扫描服务（任务）
- 第 5 章，控制对设备的访问（任务）
- 第 6 章，使用基本审计报告工具（任务）
- 第 7 章，控制对文件的访问（任务）

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“控制对计算机系统的访问”
- 第 40 页中的“控制对设备的访问”
- 第 42 页中的“控制对计算机资源的访问”
- 第 45 页中的“控制对文件的访问”
- 第 47 页中的“控制网络访问”
- 第 51 页中的“报告安全问题”

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。您负责此大型系统的安全。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。您还需要确保网络中计算机上的数据完整性。

在文件级别，Oracle Solaris 提供了可用于保护文件、目录和设备的安全功能。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 and 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 40 页中的“控制对设备的访问”
- 第 42 页中的“控制对计算机资源的访问”
- 第 45 页中的“控制对文件的访问”
- 第 47 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。入侵者可以获得操作系统和网络的访问权限。必须为计算机环境和计算机硬件提供物理保护，防止其受到未经授权的访问。

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 `eeeprom` 命令要求在访问 PROM 时提供口令。有关更多信息，请参见第 64 页中的“如何要求提供用于访问硬件的口令”。要保护 x86 硬件，请参考供应商文档。

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必须有一个口令。口令是一种简单的验证机制。如果帐户没有口令，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。

用户登录到系统时，`login` 命令会根据名称转换服务

`svc:/system/name-service/switch` 中的信息检查相应的命名服务或目录服务数据库。以下数据库可以影响登录：

- `files`—指定本地系统上的 `/etc` 文件
- `ldap`—指定 LDAP 服务器上的 LDAP 目录服务
- `nis`—指定 NIS 主服务器上的 NIS 数据库
- `dns`—指定网络上的域名服务

有关命名服务的说明，请参见 `nsd(1M)` 手册页。有关命名服务和目录服务的信息，请参见《[Oracle Solaris Administration: Naming and Directory Services](#)》。

`login` 命令可验证用户提供的用户名和口令。如果用户名不在口令数据库中，`login` 命令将拒绝对系统的访问。如果口令对指定的用户名不正确，则 `login` 命令将拒绝对系统的访问。当用户提供有效的用户名及其对应的口令时，系统将授予该用户访问系统的权限。

成功登录到系统之后，PAM 模块可以简化应用程序登录。有关更多信息，请参见第 15 章，使用 PAM。

Oracle 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 48 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，必须同时提供用户名和口令。虽然登录名是公开的，但是口令必须保密。口令应该只有相应的用户才知道。用户必须仔细选择他们的口令，且必须经常更改这些口令。

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。此外，还可以通过锁定口令来禁用用户帐户。有关口令管理的详细信息，请参见《[Oracle Solaris 管理：常见任务](#)》中的第 2 章“管理用户帐户和组（概述）”和 [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 `/etc/passwd` 文件中。加密的口令本身保存在单独的投影文件 `/etc/shadow` 中。这种安全措施可防止用户获取对加密口令的访问权限。虽然任何可以登录到系统的用户都能使用 `/etc/passwd` 文件，但是只有超级用户才能读取 `/etc/shadow` 文件。可以使用 `passwd` 命令更改本地系统上用户的口令。

NIS 口令

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

LDAP 口令

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

在 Oracle Directory Server 企业版上实施口令策略。具体而言，客户机的 `pam_ldap` 模块遵循在 Oracle Directory Server 企业版上实施的口令策略控制。有关更多信息，请参见《[Oracle Solaris Administration: Naming and Directory Services](#)》中的“LDAP Naming Services Security Model”。

口令加密

强口令加密可针对攻击提供一道前期防线。Oracle Solaris 软件提供了六种口令加密算法。与 UNIX 算法相比，[Blowfish](#)、[MD5](#) 和 SHA 算法提供的口令加密更加稳健。

口令算法标识符

可以在 `/etc/security/policy.conf` 文件中为您的站点指定算法配置。在 `policy.conf` 文件中，算法通过其标识符指定，如下表所示。有关标识符-算法映射的信息，请参见 `/etc/security/crypt.conf` 文件。

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_bsmd5(5)

表 2-1 口令加密算法 (续)

标识符	说明	算法手册页
2a	与 BSD 系统上的 Blowfish 算法兼容的 Blowfish 算法。	crypt_bsdbf(5)
md5	Sun MD5 算法，一般认为此算法比 BSD 和 Linux 版本的 MD5 更大。	crypt_sunmd5(5)
5	SHA256 算法。SHA 是 Secure Hash Algorithm（安全散列算法）的缩写。此算法属于 SHA-2 系列。SHA256 支持 255 个字符的口令。	crypt_sha256(5)
6	SHA512 算法。	crypt_sha512(5)
__unix__	传统的 UNIX 加密算法。	crypt_unix(5)

policy.conf 文件中的算法配置

下面显示了 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,md5,5,6

# 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 Oracle Solaris default is a SHA256 based algorithm. To revert to
# the policy present in Solaris releases set CRYPT_DEFAULT=__unix__,
# which is not listed in crypt.conf(4) since it is internal to libc.
#
CRYPT_DEFAULT=5
...
```

更改 CRYPT_DEFAULT 的值后，将使用与新值关联的算法对新用户的口令进行加密。

现有用户更改其口令之后，其旧口令的加密方式会影响新口令所用的加密算法。例如，假设 CRYPT_ALGORITHMS_ALLOW=1,2a,md5,5,6，CRYPT_DEFAULT=1。下表说明了生成加密口令所使用的算法。

标识符 = 口令算法		
初始口令	更改后的口令	说明
1 = crypt_bsdbf	使用同一算法	1 标识符也是 CRYPT_DEFAULT 的值。继续使用 crypt_bsdbf 算法对用户口令进行加密。

标识符 = 口令算法		
初始口令	更改后的口令	说明
2a = crypt_bsdbf	使用同一算法	2a 标识符在 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 标识符在 CRYPT_ALGORITHMS_ALLOW 列表中。因此，使用 crypt_md5 算法对新口令进行加密。
5 = crypt_sha256	使用同一算法	5 标识符在 CRYPT_ALGORITHMS_ALLOW 列表中。因此，使用 crypt_sha256 算法对新口令进行加密。
6 = crypt_sha512	使用同一算法	6 标识符在 CRYPT_ALGORITHMS_ALLOW 列表中。因此，使用 crypt_sha512 算法对新口令进行加密。
__unix__ = crypt_unix	使用 crypt_bsdbf 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 59 页中的“更改口令加密的缺省算法（任务）”。

特殊系统帐户

root 帐户是多个特殊系统帐户之一。在这些帐户中，只有 root 帐户指定了口令并可登录。nuucp 帐户可登录进行文件传输。其他系统帐户无需使用 root 的全部功能即可保护文件或运行管理进程。



注意 – 绝对不要更改系统帐户的口令设置。Oracle Solaris 中提供的系统帐户是有安全保障的。

下表列出了一些系统帐户及其用途。这些系统帐户执行一些特殊功能。每个帐户都有一个小于 100 的 UID。

表 2-2 系统帐户及其用途

系统帐户	UID	用途
root	0	几乎没有任何限制。可以覆盖其他保护和权限。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 帐户可以使用大多数 Oracle Solaris 命令。
daemon	1	控制后台处理。
bin	2	有权使用部分 Oracle Solaris 命令。
sys	3	有权访问许多系统文件。

表 2-2 系统帐户及其用途 (续)

系统帐户	UID	用途
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-to-UNIX copy program, 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。Oracle Solaris 提供多个命令用于监视、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、帧缓存器或音频设备) 的权限。有关更多信息, 请参见 [logindevperm\(4\)](#) 手册页。

控制对设备的访问

连接到计算机系统的外围设备会造成安全风险。麦克风可以截获对话内容并将其传输到远程系统。CD-ROM 可能会遗留信息, 而 CD-ROM 设备的下一个用户可以看到这些信息。可以从远程访问打印机。系统自带设备也可能会带来安全问题。例如, 网络接口 (如 bge0) 被视为自带设备。

Oracle Solaris 软件提供了两种方法用于控制对设备的访问。**设备策略**可限制或防止对系统自带设备的访问。设备策略在内核中实施。**设备分配**可限制或防止对外围设备的访问。设备分配在用户分配时实施。

设备策略使用特权在内核中保护选定的设备。例如, 对网络接口 (如 bge) 实施的设备策略要求具有所有特权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, 设备分配处于禁用状态。一旦启用, 便可配置设备分配以防止使用设备, 或者要求获得访问设备的授权。当分配某个设备供使用时, 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 Oracle Solaris 系统以控制对设备的访问：

- **设置设备策略**—在 Oracle Solaris 中，可以要求访问特定设备的进程使用一个特权集运行。没有这些特权的进程将无法使用设备。在系统引导时，Oracle 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，管理员可以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个时间只由一个用户使用。可以进一步要求用户满足某些安全要求。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个设备（如麦克风）。计算机资讯服务站可能是一个使特定设备不可用的不错选择。
- **将设备限定到特定区域**—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《Oracle Solaris 管理：Oracle Solaris Zones、Oracle Solaris 10 Zones 和资源管理》中的“非全局区域中的设备使用”。有关设备和区域的更多一般介绍，请参见《Oracle Solaris 管理：Oracle Solaris Zones、Oracle Solaris 10 Zones 和资源管理》中的“区域中的已配置设备”。

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某些特权。仅有正在使用设备策略指定的特权运行的进程才能访问受设备策略保护的设备。Oracle Solaris 提供了缺省设备策略。例如，`bge0` 之类的网络接口要求对其进行访问的进程使用 `net_rawaccess` 特权运行。该要求是在内核中实施的。有关特权的更多信息，请参见第 138 页中的“[特权（概述）](#)”。

在更早的发行版中，仅通过文件权限保护设备节点。例如，组 `sys` 拥有的设备只能由该 `sys` 组的成员打开。现在，文件权限不预先决定谁可以打开设备。取而代之的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 权限。但是，设备只能由具有相应特权的进程打开。

可以审计设备策略的配置。`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 74 页中的“[配置设备策略（任务列表）](#)”
- 第 85 页中的“[设备策略命令](#)”
- 第 144 页中的“[特权和设备](#)”

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，则仅通过文件权限保护外围设备。例如，外围设备在缺省情况下可供以下用户使用：

- 可以对磁盘或 CD-ROM 进行读写操作的任何用户。
- 可以连接麦克风的任何用户。
- 可以访问已连接的打印机的任何用户。

设备分配可以将设备限制为仅由授权用户使用。设备分配还可以完全阻止对设备的访问。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，设备清除脚本将删除所有遗留数据。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 90 页中的“编写新的设备清除脚本”。

可以对分配设备、解除分配设备以及列出可分配设备的企图进行审计。审计事件属于 other 审计类。

有关设备分配的更多信息，请参见以下内容：

- 第 77 页中的“管理设备分配（任务列表）”
- 第 85 页中的“设备分配”
- 第 86 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。您可以通过设置限制访问的对象及其可以访问的资源。可以记录资源的使用，并可以监视正在使用资源的对象，还可以设置系统以最大程度地减少资源的不当使用。

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 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 用户。可以监视一直在使用 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。有关监视超级用户以及限制超级用户访问权限的过程，请参见第 61 页中的“监视和限制超级用户（任务）”。

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 (Role-based access control, RBAC) 是 Oracle Solaris 的一项功能，目的在于将超级用户的功能分配给多个管理角色。超级用户（即 root 用户）可以访问系统中的所有资源。借助 RBAC，可以使用一组具有独立权限的角色来替换 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 127 页中的“[基于角色的访问控制（概述）](#)”。

防止无意中误用系统资源

可以通过以下方法来防止您和您的用户发生意外错误：

- 可以通过正确设置 `PATH` 变量来阻止运行特洛伊木马。
- 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

应注意正确设置 `PATH` 变量。否则，可能会无意中运行他人引入的程序。入侵程序可能会破坏数据或危害系统。这种对安全性构成威胁的程序称为**特洛伊木马**。例如，一个冒充的 `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该冒充程序。这类脚本看上去就像常规的 `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。

`PATH` 变量是在登录时自动设置的。`PATH` 是通过您的初始化文件设置的，如 `.bashrc` 和 `/etc/profile`。当设置用户搜索路径以便将当前目录（`.`）置于最后时，就可以防止您运行这类特洛伊木马。供 `root` 帐户使用的 `PATH` 变量绝不能包括当前目录。

为用户指定受限 `Shell`

标准 `shell` 允许用户执行打开文件、执行命令等操作。受限 `shell` 可限制用户更改目录和执行命令的能力。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`：

- 受限用户只能使用用户的起始目录，因此无法使用 `cd` 命令更改目录。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。
- 该类用户无法使用 `>` 或 `>>` 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。此类 `shell` 将为需要执行特定任务的用户创建一种受限环境。但是，受限 `shell` 并非完全安全，它的目的仅在于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。

限制对文件中数据的访问

因为 Oracle Solaris 是一种多用户环境，因此文件系统安全性是系统中最基本的安全风险。可以使用传统的 UNIX 文件保护措施保护文件，也可以使用更安全的访问控制列表 (access control list, ACL)。

您可能需要允许一些用户读取某些文件，为另一些用户提供更改或删除某些文件的权限。您可能会有一些不希望其他任何用户看到的数据。[第 7 章，控制对文件的访问（任务）](#)介绍了如何设置文件权限。

限制 setuid 可执行文件

可执行文件可能产生安全风险。许多可执行程序必须以 root 身份运行才能正常工作。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。如果在编写某个程序时没有注意安全问题，使用 root ID 运行该程序将造成潜在的安全问题。

除了 setuid 位设置为 root 的 Oracle 随附可执行程序外，应该禁止使用其他 setuid 程序。如果无法禁止使用 setuid 程序，则必须限制其使用。安全管理需要的 setuid 程序很少。

有关更多信息，请参见第 116 页中的“防止可执行文件危及安全”。有关过程，请参见第 122 页中的“防止程序受到安全风险（任务列表）”。

使用缺省安全 (Secure by Default) 配置

缺省情况下，安装 Oracle Solaris 后，将禁用许多网络服务。此配置称为“缺省安全”(SBD)。借助 SBD，接受网络请求的唯一网络服务将是 sshd 守护进程。其他所有网络服务将被禁用，或仅处理本地请求。要启用单个网络服务（如 ftp），可使用 Oracle Solaris 的服务管理工具 (Service Management Facility, SMF) 功能。有关更多信息，请参见 [netservices\(1M\)](#) 和 [smf\(5\)](#) 手册页。

使用资源管理功能

Oracle 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 Oracle Solaris 资源管理功能，可以为特定项目指定资源，还可以动态调整可用资源。有关更多信息，请参见《[Oracle Solaris 管理：Oracle Solaris Zones、Oracle Solaris 10 Zones 和资源管理](#)》中的第 I 部分，“Oracle Solaris 资源管理”。

使用 Oracle Solaris 区域

Oracle Solaris 区域提供了一种应用程序执行环境，其中单个 Oracle 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。

Oracle Solaris 区域非常适用于将多个应用程序置于单个服务器上的环境。有关更多信息，请参见《[Oracle Solaris 管理：Oracle Solaris Zones、Oracle Solaris 10 Zones 和资源管理](#)》中的第 II 部分，“Oracle Solaris Zones”。

监视计算机资源的使用情况

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

- 什么是正常负载？
- 谁有权访问系统？
- 个人何时访问系统？
- 哪些程序会在系统上正常运行？

了解这些知识之后，就可以使用可用工具审计系统使用情况并监视单个用户的活动。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 26 章，[审计（概述）](#)。

监视文件完整性

作为系统管理员，您需要确保所管理的系统上安装的文件未被意外更改。在大型安装中，通过有关每个系统上的软件栈的比较和报告工具可以跟踪系统。使用基本审计报告工具 (Basic Audit Reporting Tool, BART)，可以按时间进度对一个或多个系统执行文件级检查，以此进行全面的系统验证。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 6 章，[使用基本审计报告工具（任务）](#)。

控制对文件的访问

Oracle Solaris 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 7 章，[控制对文件的访问（任务）](#)。有关设置相应文件权限的逐步说明，请参见第 116 页中的“[保护文件（任务）](#)”。

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 **600** 权限的文件只能由其所有者和超级用户读取。具有 **700** 权限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该文件。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。

加密框架提供 **digest**、**mac** 和 **encrypt** 命令来保护文件。有关更多信息，请参见第 11 章，[加密框架（概述）](#)。

使用访问控制列表

ACL（发音为 ackkl）可以提供更多的文件权限控制权。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 文件保护可为以下三类用户提供读写和执行权限：所有者、组和其他用户。ACL 可提供更精细的文件安全性。

使用 ACL，您可以定义细粒度文件权限，包括以下权限：

- 所有者文件权限
- 所有者组的文件权限
- 不属于所有者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 115 页中的“使用访问控制列表保护 UFS 文件”。要使用访问控制列表 (access control list, ACL) 保护 ZFS 文件，请参见《[Oracle Solaris 管理：ZFS 文件系统](#)》中的第 8 章“使用 ACL 和属性保护 Oracle Solaris ZFS 文件”。

跨计算机共享文件

网络文件服务器可以控制哪些文件可供共享。网络文件服务器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。一般情况下，文件服务器可以为所有客户机或特定客户机授予读写访问权限或只读访问权限。通过 **share** 命令使资源可用时，将会指定访问控制。

创建某个 ZFS 文件系统的 NFS 共享后，该文件系统将永久共享，直到共享被删除为止。SMF 会在系统重新引导后自动管理共享。有关更多信息，请参见《[Oracle Solaris 管理：ZFS 文件系统](#)》中的第 3 章“Oracle Solaris ZFS 与传统文件系统之间的差别”。

限制对共享文件的 root 访问

一般情况下，不允许超级用户对网络中共享的文件系统进行 root 访问。NFS 系统将请求程序的用户更改为用户 ID 为 60001 的用户 nobody，以此防止对挂载的文件系统进行 root 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对共享文件系统的 root 访问权限。要授予这些特权，请将 share 命令与 root=hostname 选项结合使用。应该谨慎使用此选项。有关 NFS 安全选项的介绍，请参见《Oracle Solaris 管理：网络服务》中的第 6 章“访问网络文件系统（参考）”。

控制网络访问

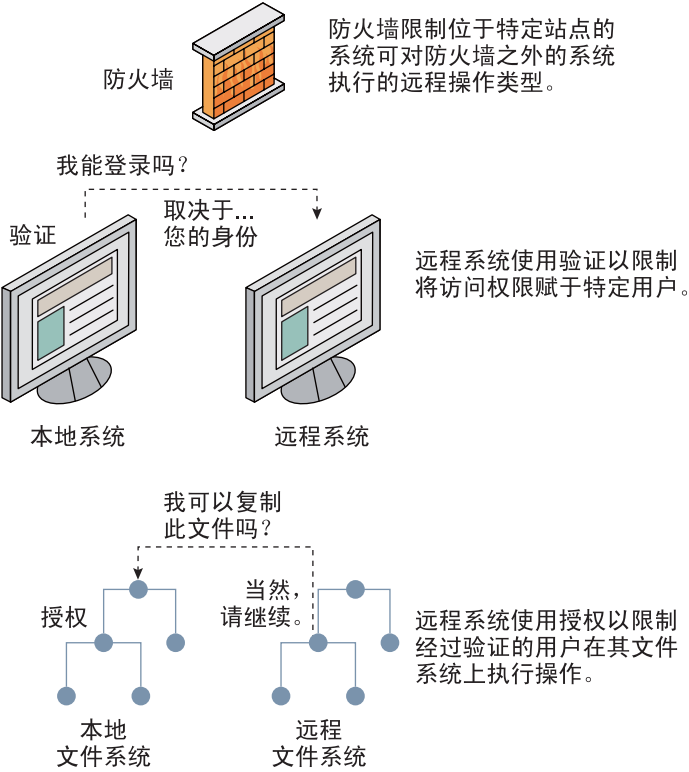
计算机常常属于某个计算机网络。连接的计算机可通过网络交换信息。联网的计算机可以访问网络中其他计算机的数据和其他资源。计算机网络营造了一种强大且完善的计算环境。但是，网络也让计算机安全问题变复杂了。

例如，在计算机网络中，单个系统允许共享信息。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而发生的访问。口令使用不当也会导致未经授权的访问。

网络安全性机制

网络安全性通常是在限制或阻止来自远程系统的操作的基础上实现的。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

验证是一种限制特定用户在访问远程系统时的访问权限的方法。可以同时¹在系统级别和网络级别设置验证。授予用户对远程系统的访问权限后，可以通过²授权来限制该用户可执行的操作。下表列出了提供验证和授权的服务。

表 2-3 远程访问的验证服务

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。	《Oracle Solaris 管理：IP 服务》 中的第 14 章“IP 安全体系结构（概述）”
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 302 页中的“Kerberos 服务的工作原理”。
LDAP	LDAP 目录服务可以在网络级别同时提供验证和授权。	《Oracle Solaris Administration: Naming and Directory Services》

表 2-3 远程访问的验证服务 (续)

服务	说明	更多信息
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。例如， <code>rlogin</code> 、 <code>rcp</code> 和 <code>ftp</code> 就是远程登录命令。如果是“可信主机”，则会自动执行验证。否则，系统会要求进行自我验证。	《Oracle Solaris 管理：网络服务》中的第 29 章“访问远程系统（任务）”
SASL	简单验证和安全层 (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 265 页中的“SASL（概述）”
安全 RPC	安全 RPC 通过对远程计算机上发出请求的用户进行验证来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 247 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 247 页中的“NFS 服务和安全 RPC”
安全 Shell	安全 Shell 对不安全网络上的网络通信流量进行加密。安全 Shell 通过使用口令和/或公钥来提供验证。安全 Shell 针对公钥使用 RSA 和 DSA 验证。	第 269 页中的“安全 Shell（概述）”

安全 RPC 的可能替代项是 Oracle 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 Oracle 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，则会显示类似如下的错误消息：

```
"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**是一种安全主机，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。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。

在某些内部网络之间也可以使用防火墙。例如，可以设置防火墙或安全网关计算机来限制包的传送。除非网关计算机是包的源地址或目标地址，否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，可以允许用包传送邮件，但不允许用包传送 telnet 或 rlogin 命令。

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实施的安全性，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。

防火墙系统不应有任何可信主机。**可信主机**是指用户无需提供口令即可通过其登录的主机。防火墙系统不应共享其任何文件系统，也不应挂载其他服务器的任何文件系统。

IPsec 和 Oracle Solaris 的 IP 过滤器功能可以提供防火墙保护。有关保护网络通信流量的更多信息，请参见《Oracle Solaris 管理：IP 服务》中的第 III 部分，“IP 安全性”。

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。通过一种称为**包粉碎**的过程，网络外部的未经授权用户可以破坏或销毁数据。

包粉碎涉及在包到达其目标之前捕获这些包。然后，入侵者在包内容中加入任意数据，并将这些包发送回其原始路线。在局域网中无法实现粉碎包，因为包会同时到达所有系统（包括服务器）。但是，可以在网关上实现包粉碎，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。窃听相关的攻击不会破坏数据的完整性。窃听者会记录对话内容供以后重放，但不会模拟用户。尽管窃听攻击不会破坏数据的完整性，但是会影响保密性。可以对通过网络上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 17 章，使用安全 Shell（任务）。
- 要对网络中的数据进行加密和验证，请参见第 19 章，Kerberos 服务介绍。
- 要对 IP 数据报进行加密，请参见《Oracle Solaris 管理：IP 服务》中的第 14 章“IP 安全体系结构（概述）”。

报告安全问题

如果遇到可疑的安全违规，可以与计算机应急响应小组/协调中心 (Computer Emergency Response Team/Coordination Center, CERT/CC) 联系。CERT/CC 是美国国防部高级研究计划署 (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。此机构可以帮助您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小组联系。有关最新的联系信息，请查阅 CERT/CC (http://www.cert.org/contact_cert/) Web 站点。

控制对系统的访问（任务）

本章介绍了用于控制可以访问 Oracle 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）”
- 第 59 页中的“更改口令加密的缺省算法（任务）”
- 第 61 页中的“监视和限制超级用户（任务）”
- 第 63 页中的“控制对系统硬件的访问（任务）”

有关系统安全的概述信息，请参见第 2 章，管理计算机安全性（概述）。

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必须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。	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 59 页中的“更改口令加密的缺省算法（任务）”
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 61 页中的“监视和限制超级用户（任务）”
禁止访问硬件设置。	禁止一般用户使用 PROM。	第 63 页中的“控制对系统硬件的访问（任务）”

保证登录和口令的安全（任务）

可以限制远程登录、要求用户使用口令并要求 root 帐户使用复杂口令。也可以监视失败的访问尝试以及临时禁止登录。

保证登录和口令的安全（任务列表）

以下任务列表列出了用于监视用户登录和禁止用户登录的过程。

任务	说明	参考
更改 root 口令。	确保 root 帐户符合口令要求。	第 54 页中的“如何更改 root 口令”
显示用户的登录状态。	列出有关用户登录帐户的详细信息，如全名和口令生命周期信息。	第 55 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 56 页中的“如何显示没有口令的用户”
临时禁止登录。	拒绝用户在系统关闭或例程时登录到计算机。	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 57 页中的“如何监视所有失败的登录尝试”

▼ 如何更改 root 口令

更改 root 口令时，必须符合适用于系统所有用户的口令要求。

开始之前 您必须是 root 角色。

- 更改口令。

```
# passwd root
New Password:
Re-enter new Password:
passwd: password successfully changed for root
```

如口令不符合要求，屏幕会显示消息。消息为提示性的。三次尝试后，必须再次运行命令以更改口令。

```
passwd: Password too short - must be at least 6 characters.
passwd: The password must contain at least 2 alphabetic character(s).
passwd: The password must contain at least 1 numeric or special character(s).
```

▼ 如何显示用户的登录状态

开始之前 您必须是 root 角色。

● 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。用逗号分隔多个登录名。

`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，也可以是命名服务的口令数据库。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 `jdoe` 的登录状态。

```
# logins -x -l jdoe
jdoe      500      staff      10      Jaylee Jaye Doe
           /home/jdoe
           /bin/bash
PS 010103 10 7 -1

jdoe      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)。
Jaylee Jaye Doe  标识注释。
/home/jdoe 标识用户的起始目录。
/bin/bash 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
- 两次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

开始之前 您必须是 root 角色。

- 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。logins 命令使用本地系统中的 passwd 数据库，除非在 nsswitch.conf 文件中指定了分布式的命名服务。

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

在系统关闭或例程期间，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。

开始之前 您必须是 root 角色。

- 1 在文本编辑器中创建 **/etc/nologin** 文件。

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

- 2 添加有关系统可用性的消息。

- 3 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。关于将系统转入单用户模式的信息，请参见《在 x86 平台上引导和关闭 Oracle Solaris》中的第 3 章“关闭系统（任务）”。

▼ 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。此过程不会从桌面登录尝试中捕获失败的登录。

开始之前 您必须是 root 角色。

- 1 在 `/var/adm` 目录中创建 `loginlog` 文件。
touch /var/adm/loginlog
- 2 在 `loginlog` 文件中，为 root 用户设置读写权限。
chmod 600 /var/adm/loginlog
- 3 在 `loginlog` 文件中，将组成员关系更改为 `sys`。
chgrp sys /var/adm/loginlog
- 4 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 2010
jdoe:/dev/pts/2:Tue Nov 4 10:21:21 2010
jdoe:/dev/pts/2:Tue Nov 4 10:21:30 2010
jdoe:/dev/pts/2:Tue Nov 4 10:21:40 2010
jdoe:/dev/pts/2:Tue Nov 4 10:21:49 2010
#
```

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

此过程在 `syslog` 文件中捕获所有失败的登录尝试。

开始之前 您必须是 root 角色。

- 1 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
# The SYSLOG_FAILED_LOGINS variable is used to determine how many failed
#SYSLOG_FAILED_LOGINS=5
SYSLOG_FAILED_LOGINS=0
#
```

- 2 使用正确的权限创建文件以保存日志记录信息。

- 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
```

- 3 编辑 **syslog.conf** 文件以记录失败的口令尝试。

将失败发送至 **authlog** 文件。

- a. 在 **syslog.conf** 文件中键入以下项。

使用制表符分隔 **syslog.conf** 中同一行上的字段。

```
auth.notice      <Press Tab> /var/adm/authlog
```

- b. 刷新 **system-log** 服务。

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

- 4 检验日志是否正常工作。

例如，使用错误的口令以一般用户的身份登录系统。然后，作为超级用户，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
#
```

- 5 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。

更改口令加密的缺省算法（任务）

缺省情况下，使用 `crypt_sha256` 算法对用户口令进行加密。可以通过更改缺省口令加密算法来使用不同的加密算法。

▼ 如何指定口令加密的算法

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 Oracle Solaris、BSD 和 Linux 版本的 UNIX 的系统构成的混合网络。有关口令加密算法和算法标识符的列表，请参见表 2-1。

开始之前 您必须是 root 角色。

- **指定所选加密算法的标识符。**

键入该标识符作为 `/etc/security/policy.conf` 文件中 `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,md5,5,6
#
# Use the version of MD5 (5) that works with Linux and BSD systems.
# Passwords previously encrypted with SHA256 (1) will be encrypted
# with MD5 when users change their passwords.
#
#
#CRYPT_DEFAULT=5
CRYPT_DEFAULT=1
```

在此示例中，算法配置可确保不使用算法 `sha256` 对口令进行加密。使用 `sha256` 模块加密口令的用户将在更改其口令时获得使用 `crypt_bsmd5` 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 在异构环境中限制口令加密算法

此示例中，包含 BSD 和 Linux 系统的网络上的管理员将口令配置为适用于所有系统。由于某些网络应用程序无法处理 SHA512 加密，因此管理员不将其标识符包含在允许的算法列表中。管理员保留 SHA256 算法，5 作为 CRYPT_DEFAULT 变量的值。CRYPT_ALGORITHMS_ALLOW 变量包含与 BSD 和 Linux 系统兼容的 MD5 标识符，以及与 BSD 系统兼容的 Blowfish 标识符。由于 5 是 CRYPT_DEFAULT 算法，因此无需在 CRYPT_ALGORITHMS_ALLOW 列表中列出。不过，为便于维护，管理员将 5 放置在了 CRYPT_ALGORITHMS_ALLOW 列表中，将未使用的标识符放置在了 CRYPT_ALGORITHMS_DEPRECATED 列表中。

```
CRYPT_ALGORITHMS_ALLOW=1,2a,5
#CRYPT_ALGORITHMS_DEPRECATED=__unix__,md5,6
CRYPT_DEFAULT=5
```

▼ 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 客户机将查阅 /etc/security/policy.conf 文件中的本地算法配置。NIS 客户机系统将对口令进行加密。

开始之前 您必须是 root 角色。

- 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 域中的每个客户机系统。
- 3 为最大限度地减少混淆，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 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。

开始之前 您必须是 root 角色。

- 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 域中的每个客户机系统。

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 `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对口令进行加密。PAM 项还允许对口令进行验证。

当 LDAP 域中的用户更改其口令时，LDAP 客户机会查阅 `/etc/security/policy.conf` 文件中的本地算法配置。LDAP 客户机系统将对口令进行加密。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，客户机将存储的口令与其刚根据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。然后，通过使用 Oracle Directory Server 企业版的加密机制，在服务器上加密口令。要了解相关过程，请参见《[Oracle Solaris Administration: Naming and Directory Services](#)》中的第 11 章“[Setting Up Oracle Directory Server Enterprise Edition With LDAP Clients \(Tasks\)](#)”。

监视和限制超级用户（任务）

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 (role-based access control, RBAC)。有关 RBAC 的概述信息，请参见第 127 页中的“[基于角色的访问控制（概述）](#)”。要设置 RBAC，请参见第 9 章，[使用基于角色的访问控制（任务）](#)。

▼ 如何监视正在使用 `su` 命令的用户

`su`log 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。

开始之前 您必须是 `root` 角色。

- 定期监视 `/var/adm/su`log 文件的内容。

```
# more /var/adm/su
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 jdoe-root
SU 01/12 14:57 + pts/0 jdoe-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` 日志记录：

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

故障排除 包含 ??? 的项表示无法识别 `su` 命令的控制终端。通常，在显示桌面之前对 `su` 命令的系统调用包括 ???，如 `SU 10/10 08:08 + ??? root-root` 中所示。在用户启动桌面会话后，`ttynam` 命令会将控制终端的值返回至 `sulog`：`SU 10/10 10:10 + pts/3 jdoe-root`。

类似如下的项表示未在命令行中调用 `su` 命令：`SU 10/10 10:20 + ??? root-oracle`。Trusted Extensions 用户可能已使用 GUI 切换至 `oracle` 角色。

▼ 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 `root` 用户尝试。

开始之前 您必须是 `root` 角色。

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` 是否无法远程登录。

在远程系统上，尝试以 `root` 用户身份登录。

```
mach2 % ssh -l root mach1
Password: <Type root password of mach1>
Password:
Password:
```

```
Permission denied (gssapi-keyex,gssapi-with-mic,publickey,keyboard-interactive).
```

在缺省配置中，`root` 是一个角色，而角色无法登录。此外，在缺省配置中，`ssh` 协议阻止 `root` 用户登录。

3 监视成为 root 用户的尝试。

缺省情况下，会通过 SYSLOG 实用程序在控制台上显示试图成为 root 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:      <Type root password>
#
```

将在终端控制台上显示一条消息。

```
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，将在终端控制台上显示该尝试。

```
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。

控制对系统硬件的访问（任务）

可以通过要求在访问硬件设置时提供口令来保护物理系统。也可以通过禁止用户使用异常中止序列离开窗口系统来保护系统。

要保护 BIOS，请参考供应商文档。

▼ 如何要求提供用于访问硬件的口令

开始之前 必须为您分配 Device Security（设备安全），Maintenance and Repair（维护和修复）或 System Administrator（系统管理员）权限配置文件。

- 1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

- 2 在终端窗口中，键入 PROM 安全模式。

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

Changing PROM password:

New password: <Type password>

Retype new password: <Retype password>

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 可选要更改 PROM 口令，请键入以下命令：

```
# eeprom security-password= Press Return
```

Changing PROM password:

New password: <Type password>

Retype new password: <Retype password>

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注意到这些情况。



注意 – 切勿忘记 PROM 口令。如果没有此口令，硬件将不可用。

▼ 如何禁用系统的异常中止序列

注 – 某些服务器系统具有钥控开关。如果将钥控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。

开始之前 您必须是 `root` 角色。

1 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
```

2 更新键盘缺省值。

```
# kbd -i
```


病毒扫描服务（任务）

本章提供了有关使用防病毒软件的信息，其中包括以下主题：

- 第 67 页中的“关于病毒扫描”
- 第 68 页中的“关于 Vscan 服务”
- 第 68 页中的“使用 Vscan 服务（任务）”

关于病毒扫描

通过称为 vscan 的扫描服务（该服务使用多种**扫描引擎**），可以保护数据不受病毒侵害。**scan engine**（**扫描引擎**）是第三方应用程序，它驻留于外部主机上，可检查文件中是否存在已知病毒。如果文件系统支持 vscan 服务且该服务已启用，并且文件的类型未被排除在外，将会对文件执行病毒扫描。如文件之前未曾针对当前病毒定义扫描，或文件自上次扫描之后做过修改，则在打开和关闭文件期间，会对文件进行病毒扫描。

可将 vscan 服务配置为使用多个扫描引擎。建议 vscan 服务使用至少两个扫描引擎。病毒扫描请求在所有可用扫描引擎上进行分布式处理。**表 4-1** 显示了支持的扫描引擎（配置有其最新修补程序时）。

表 4-1 防病毒扫描引擎软件

防病毒软件	ICAP 支持
Symantec 防病毒扫描引擎 4.3	支持
Symantec 防病毒扫描引擎 5.1	支持
Computer Associates eTrust AntiVirus 7.1	不支持 ¹
Computer Associates Integrated Threat Management 8.1	
Trend Micro InterScan Web Security Suite (IWSS) 2.5	支持

¹ 要求为 Computer Associates 防病毒扫描引擎安装 Sun StorageTek 5000 NASICAP 服务器从 **Sun Download Center**: (<http://www.oracle.com/technetwork/indexes/downloads/index.html>) 获取软件包。

表 4-1 防病毒扫描引擎软件 (续)

防病毒软件	ICAP 支持
McAfee Secure Internet Gateway 4.5	支持

关于 Vscan 服务

即时扫描方法的优势在于：可在使用文件之前使用最新病毒定义对其进行扫描。通过使用此方法，可在病毒危害数据前检测到它们。

下面描述了病毒扫描过程：

1. 在用户打开客户机上的文件时，vscan 服务根据文件是否之前已使用最新病毒定义进行过扫描以及文件自上次扫描后是否修改过而决定是否需要对文件进行扫描。
 - 如需要扫描，文件将会传输至scan engine（扫描引擎）。如果无法连接至某个扫描引擎，文件将会传输另一个扫描引擎。如果无扫描引擎可用，则无法执行病毒扫描，对文件的访问可能被拒绝。
 - 如果文件无需扫描，则允许客户机访问此文件。
2. 扫描引擎使用最新病毒定义对文件进行扫描。
 - 如果检测到病毒，文件将被标记为隔离状态。被隔离的文件不能读取，执行或者重命名，但可以删除。系统日志会记录被隔离文件的文件名和病毒名称，如果已启用审计，还会创建含有相同信息的审计记录。
 - 如文件未受感染，将会在文件上标记一个扫描戳记，并允许客户机访问文件。

使用 Vscan 服务（任务）

满足下列要求时，可对文件进行病毒扫描：

- 至少安装并配置一个扫描引擎。
- 文件驻留的文件系统支持病毒扫描。
- 已在文件系统上启用病毒扫描。
- 已启用 vscan 服务。
- 已将 vscan 服务配置为扫描指定文件类型的文件。

下表列出了设置 vscan 服务需要执行的任务。

任务	说明	参考
安装scan engine（扫描引擎）。	安装并配置表 4-1 中列出的一个或多个受支持的第三方产品。	请参见产品文档。
启用文件系统以允许病毒扫描。	启用 ZFS 文件系统上的病毒扫描。缺省情况下，扫描处于禁用状态。	第 69 页中的“如何在文件系统上启用病毒扫描”

任务	说明	参考
启用 vscan 服务。	开始扫描服务。	第 69 页中的“如何启用 Vscan 服务”
将扫描引擎添加至 vscan 服务。	在 vscan 服务中包括特定扫描引擎。	第 70 页中的“如何添加扫描引擎”
配置 vscan 服务。	查看并更改 vscan 属性。	第 70 页中的“如何查看 Vscan 属性” 第 70 页中的“如何更改 Vscan 属性”
针对特定文件类型配置 vscan 服务。	指定要在扫描中包括和排除的文件类型。	第 71 页中的“如何从病毒扫描操作中排除文件”

▼ 如何在文件系统中启用病毒扫描

使用文件系统命令允许对文件进行病毒扫描。例如，要将 ZFS 文件系统包括在病毒扫描中，可使用 `zfs(1M)` 命令。

开始之前 您必须分配有 ZFS File System Management（ZFS 文件系统管理）或 ZFS Storage Management（ZFS 存储管理）权限配置文件。ZFS 文件系统允许将部分管理任务委托给特定用户。有关委托管理的更多信息，请参见《Oracle Solaris 管理：ZFS 文件系统》中的第 9 章“Oracle Solaris ZFS 委托管理”。

- 1 成为具有所需安全属性的管理员。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。
- 2 在 ZFS 文件系统（如 `pool/volumes/vol1`）上启用病毒扫描。
`# zfs set vscan=on path/pool/volumes/vol1`

▼ 如何启用 Vscan 服务

开始之前 您必须分配有 VSCAN Management（VSCAN 管理）权限配置文件。

- 1 成为具有所需安全属性的管理员。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。
- 2 使用 `svcadm(1M)` 命令启用病毒扫描。
`# svcadm enable vscan`

▼ 如何添加扫描引擎

开始之前 您必须分配有 VSCAN Management（VSCAN 管理）权限配置文件。

- 1 成为具有所需安全属性的管理员。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。
- 2 要将扫描引擎添加到 **vscan** 服务，并保留缺省属性，请键入：

```
#vscanadm add-engine engine_ID
```


有关命令说明，请参见vscanadm(1M) 命令手册页。

▼ 如何查看 Vscan 属性

开始之前 您必须分配有 VSCAN Management（VSCAN 管理）权限配置文件。

- 1 成为具有所需安全属性的管理员。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。
- 2 查看 **vscan** 服务的属性（所有扫描引擎或部分扫描引擎）。
 - 要查看特定扫描引擎的属性，请键入：

```
# vscanadm get-engine engineID
```
 - 要查看所有扫描引擎的属性，请键入：

```
# vscanadm get-engine
```
 - 要查看某一 **vscan** 服务属性，请键入：

```
# vscanadm get -p property
```


其中 *property* 是 vscanadm(1M) 命令手册页中描述的参数之一。
例如，如果想查看可扫描的文件的最大大小，请键入：

```
# vscanadm get max-size
```

▼ 如何更改 Vscan 属性

可以更改特定扫描引擎的属性和 **vscan** 服务的一般属性。许多扫描引擎限制所扫描文件的大小，因此为 **vscan** 服务的 *max-size* 属性设置的值必须小于或等于扫描引擎允许的最大值。然后，可以定义因超过最大大小而未被扫描的文件是否可以访问。

开始之前 您必须分配有 VSCAN Management（VSCAN 管理）权限配置文件。

- 1 成为具有所需安全属性的管理员。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。
- 2 使用 `vscanadm show` 命令查看当前属性。
- 3 设置病毒扫描的最大大小，例如 128 MB。
`# vscanadm set -p max-size=128M`
- 4 指定拒绝访问由于其他大小而未被扫描的所有文件。
`# vscanadm set -p max-size-action=deny`
有关命令说明，请参见 `vscanadm(1M)` 命令手册页。

▼ 如何从病毒扫描操作中排除文件

当启用防病毒保护时，可以指定将特定类型的所有文件从病毒扫描中排除。由于 `vscan` 服务会影响系统性能，您可以通过指定针对特定的文件类型进行病毒扫描来节省系统资源。

开始之前 您必须分配有 VSCAN Management（VSCAN 管理）权限配置文件。

- 1 成为具有所需安全属性的管理员。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。
- 2 查看病毒扫描中包含的所有文件类型的列表。
`# vscanadm get -p types`
- 3 指定要进行病毒扫描的文件类型：
 - 从病毒扫描操作中排除特定文件类型，例如 JPEG 类型。
`# vscanadm set -p types=-jpg, +*`
 - 在病毒扫描操作中包括特定文件类型，例如可执行文件。
`# vscanadm set -p types=+exe, -*`
 有关更多信息，请参见 `vscanadm(1M)` 手册页。

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。

以下是本章中信息的列表。

- 第 73 页中的“配置设备（任务列表）”
- 第 74 页中的“配置设备策略（任务）”
- 第 76 页中的“管理设备分配（任务）”
- 第 81 页中的“分配设备（任务）”
- 第 84 页中的“设备保护（参考）”

有关设备保护的概述信息，请参见第 40 页中的“控制对设备的访问”。

配置设备（任务列表）

以下任务列表列出了要执行以便管理对设备的访问的任务。

任务	参考
管理设备策略。	第 74 页中的“配置设备策略（任务列表）”
管理设备分配。	第 77 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）”

配置设备策略（任务）

设备策略可限制或禁止访问属于系统一部分的设备。策略是在内核中实施的。

配置设备策略（任务列表）

以下任务列表列出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 74 页中的“如何查看设备策略”
要求使用设备的特权。	使用特权保护设备。	第 75 页中的“如何更改现有设备上的设备策略”
删除设备的特权要求。	删除或减少访问设备所需的特权。	示例 5-3
审计设备策略的更改。	在审计迹中记录设备策略的更改。	第 76 页中的“如何审计设备策略的更改”
访问 /dev/arp。	获取 Oracle Solaris IP MIB-II 信息。	第 76 页中的“如何从 /dev/* 设备检索 IP MIB-II 信息”

▼ 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 5-1 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/bge
/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/bge
read_priv_set=net_rawaccess
write_priv_set=net_rawaccess
```

▼ 如何更改现有设备上的设备策略

开始之前 您必须分配有 Device Security（设备安全）权限配置文件。

1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

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。设备策略指定两个特权集。一个用于读取设备，另一个用于写入设备。

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 5-2 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 5-3 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特权集。

```
# 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
```

▼ 如何审计设备策略的更改

缺省情况下，as 审计类包括 AUE_MODDEVPLCY 审计事件。

开始之前 您必须分配有 Audit Configuration（审计配置）权限配置文件。

- 1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

- 2 预选包括 AUE_MODDEVPLCY 审计事件的审计类。

```
# auditconfig -getflags  
current-flags  
# auditconfig -setflags current-flags,as
```

有关详细说明，请参见第 504 页中的“如何预选审计类”。

▼ 如何从 /dev/* 设备检索 IP MIB-II 信息

检索 Oracle 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 特权，而 /dev/arp 不需要特权。

- 2 打开 /dev/arp 并推送 tcp 和 udp 模块。

此方法不需要特权，它等同于打开 /dev/ip 并推送 arp、tcp 和 udp 模块。现在，由于打开 /dev/ip 需要特权，因此 /dev/arp 是首选方法。

管理设备分配（任务）

设备分配会限制或禁止访问外围设备。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。

管理设备分配（任务列表）

以下任务列表列出了启用和配置设备分配的过程。缺省情况下，设备分配处于禁用状态。启用设备分配后，请参见第 81 页中的“分配设备（任务）”以获取有关分配设备的说明。

任务	说明	参考
使设备可分配。	使设备可以一次分配给一个用户。	第 77 页中的“如何启用设备分配”
禁用设备分配。	删除所有设备中的分配限制。	
授予用户分配设备的授权。	向用户指定设备分配授权。	第 78 页中的“如何授权用户分配设备”
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 79 页中的“强制分配设备”
强制解除设备分配。	解除当前分配给用户的设备的分配。	第 79 页中的“强制解除分配设备”
更改设备的分配属性。	更改分配设备的要求。	第 80 页中的“如何更改可分配的设备”
创建设备清除脚本。	清除物理设备中的数据。	第 90 页中的“编写新的设备清除脚本”
审计设备分配	在审计迹中记录设备分配	第 81 页中的“如何审计设备分配”

▼ 如何启用设备分配

开始之前 您必须分配有 Device Security（设备安全）权限配置文件。

- 1 成为具有所需安全属性的管理员。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。
- 2 启用设备分配服务，并验证是否已启用此服务。

```
# svcadm enable svc:/system/device/allocate
# svcs -x allocate
svc:/system/device/allocate:default (device allocation)
State: online since September 10, 2011 01:10:11 PM PDT
See: allocate(1)
See: deallocate(1)
See: list_devices(1)
See: device_allocate(1M)
See: mkdevalloc(1M)
See: mkdevmaps(1M)
See: dminfo(1M)
See: device_maps(4)
See: /var/svc/log/system-device-allocate:default.log
Impact: None.
```

要禁用设备分配服务，请使用 `disable` 子命令。

```
# svcadm disable device/allocate
```

▼ 如何授权用户分配设备

开始之前 您必须分配有 User Security（用户安全）权限配置文件。

1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“[如何获取管理权限](#)”。

2 创建包含适当授权和命令的权限配置文件。

通常，您会创建一个包括 `solaris.device.allocate` 授权的权限配置文件。请按第 159 页中的“[如何创建或更改权限配置文件](#)”中的说明操作。为权限配置文件指定相应的属性，例如：

- 权限配置文件名称：Device Allocation（设备分配）
- 授予的授权：`solaris.device.allocate`
- 具有安全属性的命令：在 `exec_attr` 数据库中，带 `sys_mount` 特权的 `mount` 和带 `sys_mount` 特权的 `umount`

3 创建权限配置文件的角色。

请按第 155 页中的“[如何创建角色](#)”中的说明操作。使用以下角色属性作为指南：

- 角色名称：`devicealloc`
- 角色全名：Device Allocator（设备分配器）
- 角色说明：Allocates and mounts allocated devices（分配并挂载已分配的设备）
- 权限配置文件：Device Allocation（设备分配）
此权限配置文件必须是该角色包含的配置文件列表中的第一个。

4 将角色指定给允许分配设备的每个用户。

5 向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 81 页中的“[如何分配设备](#)”。

▼ 如何查看有关设备的分配信息

开始之前 已完成第 77 页中的“[如何启用设备分配](#)”。

您必须分配有 Device Security（设备安全）权限配置文件。

- 1 成为具有所需安全属性的管理员。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。

- 2 显示有关您系统上的可分配设备的信息。

```
# 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]*—磁盘驱动器。
- *rmdisk[n]*—是可移除介质设备。
- *sr[n]*—CD-ROM 驱动器。
- *st[n]*—磁带机。

故障排除 如果 `list_devices` 命令返回了类似以下内容的错误消息，则表示设备分配未启用，或者您没有足够的权限来检索信息。

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

要使命令成功执行，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

可在某个用户忘记解除分配设备时使用强制分配，也可在用户即时需要设备时使用强制分配。

开始之前 您必须分配有 `solaris.device.revoke` 授权。

- 1 确定您在您的角色中是否具有相应授权。

```
$ auths
solaris.device.allocate solaris.device.revoke
```

- 2 将设备强制分配给需要设备的用户。

在此示例中，将磁带机强制分配给了用户 `jdoe`。

```
$ allocate -U jdoe
```

▼ 强制解除分配设备

进程终止或用户注销之后，不会自动解除分配用户已分配的设备。用户忘记解除分配设备时，应使用强制解除分配。

开始之前 您必须分配有 `solaris.device.revoke` 授权。

- 1 确定您在您的角色中是否具有相应授权。

```
$ auths
solaris.device.allocate solaris.device.revoke
```

- 2 强制解除分配设备。

在此示例中，强制解除分配了打印机。现在，打印机可供其他用户分配了。

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

▼ 如何更改可分配的设备

开始之前 必须启用设备分配，此过程才会成功。要启用设备分配，请参见第 77 页中的“如何启用设备分配”。您必须是超级用户。

- 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 5-4 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任何设备。`device_allocate` 文件中每个设备项的第五个字段都已更改为一个 `at` 符号 (`@`)。

```
# 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 5-5 禁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已更改为一个星号 (`*`)。

```
# 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 5-6 禁止使用所有外围设备

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

```
# 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，设备分配命令在 `other` 审计类中。

开始之前 您必须分配有 Audit Configuration（审计配置）权限配置文件。

1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 预选 `ot` 审计类。

```
# auditconfig -getflags
current-flags
# auditconfig -setflags current-flags,ot
```

有关详细说明，请参见第 504 页中的“如何预选审计类”。

分配设备（任务）

设备分配使设备每次只能由一个用户使用。必须挂载需要挂载点的设备。以下过程向用户显示了如何分配设备。

▼ 如何分配设备

开始之前 必须启用设备分配，如第 77 页中的“如何启用设备分配”中所述。如果需要授权，则用户必须具有授权。

1 分配设备。

按设备名称指定设备。

```
% allocate device-name
```

2 验证设备是否已分配。

运行相同命令。

```
% allocate device-name
allocate. Device already allocated.
```

示例 5-7 分配麦克风

在此示例中，用户 `jdoe` 将分配麦克风 `audio`。

```
% whoami
jdoe
% allocate audio
```

示例 5-8 分配打印机

在此示例中，某个用户将分配打印机。在用户解除分配打印机或强制将打印机分配给其他用户之前，其他任何用户都不能打印到 `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“强制解除分配设备”。

示例 5-9 分配磁带机

在此示例中，用户 `jdoe` 分配了磁带机 `st0`。

```
% whoami
jdoe
% allocate st0
```

故障排除 如果 `allocate` 命令不能分配设备，控制台窗口中将显示一条错误消息。有关分配错误消息的列表，请参见 [allocate\(1\)](#) 手册页。

▼ 如何挂载分配的设备

如果被授予相应的特权，则将自动挂载设备。如果设备无法挂载，请遵循以下过程。

开始之前 您已分配设备。您分配有挂载设备所需的特权。要指定所需的特权，请参见第 78 页中的“如何授权用户分配设备”。

1 承担可以分配和挂载设备的角色。

```
% su - role-name
Password: <Type role-name password>
$
```

2 在角色起始目录中创建并保护挂载点。

仅当首次需要挂载点时，才需要执行此步骤。

```
$ 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。

按设备名称指定设备。

```
$ allocate device-name
```

5 挂载设备。

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

其中

-o ro 指示将设备挂载为只读。使用 **-o rw** 指示可以写入设备。

-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。磁盘通常使用 PCFS 文件系统格式化。

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 5-10 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: <Type devicealloc password>
$ 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，则将显示一条错误消息：`mount: insufficient privileges`。检查以下各项：

- 验证是否在配置文件 `shell` 中执行 `mount` 命令。如果您已经承担了某个角色，此角色将有一个配置文件 `shell`。如果指定给您的配置文件包含 `mount` 命令，则必须创建一个配置文件 `shell`。有关可用配置文件 `shell` 的列表，请参见 [pfexec\(1\)](#)。
- 验证是否拥有指定的挂载点。您必须具有对挂载点的读写和执行访问权限。

如果仍然不能挂载分配的设备，请与管理员联系。

▼ 如何解除分配设备

通过解除分配，其他用户便可在您用完设备之后分配和使用设备。

开始之前 您必须已经分配设备。

- 1 如果已挂载设备，请取消挂载。

```
$ cd $HOME
$ umount mount-point
```

- 2 对设备解除分配。

```
$ deallocate device-name
```

示例 5-11 解除分配麦克风

在此示例中，用户 `jdoe` 将解除分配麦克风 `audio`。

```
% whoami
jdoe
% deallocate audio0
```

示例 5-12 解除分配 CD-ROM 驱动器

在此示例中，Device Allocator（设备分配器）角色将解除分配 CD-ROM 驱动器。显示消息后，会弹出 CD-ROM。

```
$ whoami
devicealloc
$ cd /home/devicealloc
$ umount /home/devicealloc/mymnt
$ ls /home/devicealloc/mymnt
$
$ deallocate sr0
/dev/sr0:      3260
/dev/rsr0:     3260
...
sr_clean: Media in sr0 is ready. Please, label and store safely.
```

设备保护（参考）

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

设备策略命令

设备管理命令可管理本地文件上的设备策略。设备策略可以包括特权要求。分配有 Device Management（设备管理）和 Device Security（设备安全）权限配置文件的用户可以管理设备。

下表列出了设备管理命令。

表 5-1 设备管理命令

命令的手册页	目的
devfsadm(1M)	管理正在运行的系统上的设备和设备驱动程序。还可装入设备策略。 使用 <code>devfsadm</code> 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 <code>/dev</code> 链接。还可以重新配置指定驱动程序的设备。
getdevpolicy(1M)	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。
add_drv(1M)	将新设备驱动程序添加到正在运行的系统。包含用于将设备策略添加到新设备的选项。通常，安装设备驱动程序时会在脚本中调用此命令。
update_drv(1M)	更新现有设备驱动程序的属性。包含用于更新设备的设备策略的选项。通常，安装设备驱动程序时会在脚本中调用此命令。
rem_drv(1M)	删除设备或设备驱动程序。

设备分配

设备分配可以保护站点，避免数据丢失、计算机病毒以及其他安全违规问题。与设备策略不同，设备分配是可选的。设备分配使用授权来限制对可分配设备的访问。

设备分配的组成部分

设备分配机制的组成部分如下所述：

- `svc:/system/device/allocate` 服务。有关更多信息，请参见 [smf\(5\)](#) 手册页和有关设备分配命令的手册页。
- `allocate`、`deallocate`、`dminfo` 和 `list_devices` 命令。有关更多信息，请参见第 86 页中的“设备分配命令”。
- Device Management（设备管理）和 Device Security（设备安全）权限配置文件。有关更多信息，请参见第 86 页中的“设备分配权限配置文件”。
- 每个可分配设备的设备清除脚本。

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` 目录中与每个可分配设备对应的锁定文件。
- 与每个可分配设备关联的锁定文件的已更改属性。

注 - 将来的 Oracle Solaris 发行版可能不支持 `/etc/security/dev` 目录。

设备分配服务

`svc:/system/device/allocate` 服务控制设备分配。缺省情况下，此服务处于禁用状态。要启用此服务，请运行 `svcadm enable svc:/system/device/allocate` 命令。

设备分配权限配置文件

需要 Device Management（设备管理）和 Device Security（设备安全）权限配置文件来管理设备和设备分配。

这些权限配置文件包括以下授权：

- `solaris.device.allocate`—分配设备所需的授权
- `solaris.device.cdrw`—读取和写入 CD-ROM 所需的授权
- `solaris.device.config`—配置设备属性所需的授权
- `solaris.device.grant`—将指定给您的设备授权委托给其他用户所需的授权
- `solaris.device.mount.alloptions.fixed`—挂载固定设备时指定挂载选项所需的授权
- `solaris.device.mount.alloptions.removable`—挂载可移除设备时指定挂载选项所需的授权
- `solaris.device.mount.fixed`—挂载固定设备所需的授权
- `solaris.device.mount.removable`—挂载可移除设备所需的授权
- `solaris.device.revoke`—撤销或回收设备所需的授权

设备分配命令

与大写的选项一起使用时，`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 5-2 设备分配命令

命令的手册页	目的
<code>dminfo(1M)</code>	根据设备类型、设备名称以及全路径名搜索可分配设备。
<code>list_devices(1)</code>	列出可分配设备的状态。 列出与 <code>device_maps</code> 文件中列出的任何设备关联的所有设备专用文件。 使用 <code>-U</code> 选项，列出可分配的设备或分配给指定用户 ID 的设备。通过此选项，可以检查哪些设备是可分配的，或者哪些设备已分配给其他用户。您必须具有 <code>solaris.device.revoke</code> 授权。
<code>allocate(1)</code>	保留一个可分配设备以供某个用户使用。 缺省情况下，用户必须拥有 <code>solaris.device.allocate</code> 授权才能分配设备。可以修改 <code>device_allocate</code> 文件以便不需要用户授权。随后，系统上的任何用户都可以请求分配设备以供使用。
<code>deallocate(1)</code>	删除设备的分配保留。

分配命令的授权

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。要创建包括 `solaris.device.allocate` 授权的权限配置文件，请参见第 78 页中的“如何授权用户分配设备”。

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 192 页中的“要求授权的选定命令”。

分配错误状态

如果 `deallocate` 命令解除分配失败，或 `allocate` 命令分配失败，设备将处于分配错误状态。如果某个可分配设备处于分配错误状态，必须强制解除分配此设备。只有拥有设备管理权限配置文件或设备安全权限配置文件的用户或角色才能处理分配错误状态。

`deallocate` 命令带 `-F` 选项时可强制解除分配。或者，可以使用 `allocate -U` 将设备指定给某个用户。分配设备之后，便可对出现的任何错误消息进行检查。更正设备的所有问题后，可对其强制解除分配。

device_maps 文件

设置设备分配时会创建设备映射。`/etc/security/device_maps` 文件包括与每个可分配设备关联的设备名称、设备类型和设备专用文件。

`device_maps` 文件定义每个设备的设备专用文件映射，这在许多情况下并不直观。使用此文件，程序可以发现设备专用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、设备类型和设备专用文件。`dminfo` 命令使用 `device_maps` 文件报告此信息。

每个设备由占一行的项表示，该项的格式如下：

device-name:device-type:device-list

示例 5-13 `device_maps` 项样例

以下是 `device_maps` 文件中磁盘驱动器 `fd0` 的一个项的样例：

```
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 (\) 结尾，后续的项另起一行。还可以包括注释。井号 (#) 对所有后续文本进行注释，直到遇到下一个前面未紧接反斜杠的新行。允许在任何字段中使用前导空格和结尾空格。字段定义如下：

- device-name* 指定设备的名称。有关当前设备名称的列表，请参见第 78 页中的“[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`、`rmdisk` 或 `audio`。*device-type* 字段按逻辑将相关设备分组。
- device-list* 列出与物理设备关联的设备专用文件。*device-list* 必须包含所有允许访问特定设备的专用文件。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* 字段的有效项反映了 `/dev` 目录中的设备文件。

device_allocate 文件

您可以修改 `/etc/security/device_allocate` 文件来将可分配设备更改为不可分配设备或添加新设备。以下是一个 `device_allocate` 样例文件。

```
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` 文件中有某个项并不表示该设备可分配，除非该项专门声明该设备可分配。在 `device_allocate` 样例文件中，可以看到音频设备项的第五个字段中使用了星号 (*)。在第五个字段中使用星号可让系统知道此设备不可分配。因此，不能使用此设备。如果此字段中存在其他值或没有值，则表示可以使用此设备。

在 `device_allocate` 文件中，每个设备由占一行的项表示，其格式如下：

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

`device_allocate` 文件中的行可以用反斜杠 (\) 结尾，后续的项另起一行。还可以包括注释。井号 (#) 对所有后续文本进行注释，直到遇到下一个前面未紧接反斜杠的新行。允许在任何字段中使用前导空格和结尾空格。字段定义如下：

<i>device-name</i>	指定设备的名称。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 ”。
<i>device-type</i>	指定通用设备类型。通用名称是设备类的名称，例如 <code>st</code> 、 <code>fd</code> 和 <code>sr</code> 。 <i>device-type</i> 字段按逻辑将相关设备分组。使某个设备可分配后，可在 <code>device_maps</code> 文件的 <i>device-type</i> 字段中检索设备名称。
<i>reserved</i>	Oracle 保留两个标记为 <code>reserved</code> 的字段供将来使用。
<i>auths</i>	指定设备是否可分配。此字段中的星号 (*) 表示此设备不可分配。授权字符串或空字段表示此设备可分配。例如， <i>auths</i> 字段中的字符串 <code>solaris.device.allocate</code> 表示需要 <code>solaris.device.allocate</code> 授权才能分配此设备。此文件中的 <code>@</code> 符号表示任何用户都可以分配此设备。
<i>device-exec</i>	提供为特殊处理（例如分配过程中的清除和对象重用保护）而调用的脚本的路径名称。每当 <code>deallocate</code> 命令对设备执行操作时，都会运行 <i>device-exec</i> 脚本。

例如，`sr0` 设备的以下项表示具有 `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定义的特征。安装新设备后，可以修改这些项。必须在设备的系统的 `device_allocate` 和 `device_maps` 文件中定义需要在使用之前分配的任何设备。目前，盒式磁带机、磁盘驱动器、CD-ROM 驱动器、可移除介质设备和音频芯片是可分配的。这些设备类型具有设备清除脚本。

注 - Xylogics 磁带机或归档磁带机也使用为 SCSI 设备提供的 `st_clean` 脚本。需要为其他设备（如终端、图形输入板和其他可分配设备）创建您自己的设备清除脚本。脚本必须满足此设备类型的对象重用要求。

设备清除脚本

通过设备分配，可以在一定程度上满足对象重用要求。**设备清除脚本**可解决安全要求，即重用某个物理设备之前从该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、磁盘驱动器、CD-ROM 驱动器和音频设备需要设备清除脚本。Oracle Solaris 提供这些脚本。本节介绍设备清除脚本的功能。

磁带的设备清除脚本

`st_clean` 设备清除脚本支持三种磁带设备：

- SCSI ¼ 英寸磁带
- 归档 ¼ 英寸磁带
- 开盘式 ½ 英寸磁带

`st_clean` 脚本使用 `mt` 命令的 `rewoffl` 选项来清除设备。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。含有介质的 ¼ 英寸磁带设备会被置于分配错误状态。分配错误状态会强制管理员手动清除设备。

正常系统操作期间，在交互模式下执行 `deallocate` 命令时，系统将提示用户删除介质。从设备中删除介质之后才会解除分配。

磁盘驱动器和 CD-ROM 驱动器的设备清除脚本

为磁盘驱动器和 CD-ROM 驱动器提供了以下设备清除脚本：

- `fd_clean` 脚本—用于磁盘的设备清除脚本。
- `sr_clean` 脚本—用于 CD-ROM 驱动器的设备清除脚本。

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。

音频的设备清除脚本

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。

编写新的设备清除脚本

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除脚本。`deallocate` 命令将参数传递到设备清除脚本中。此处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
```

设备清除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。

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的清单，以及如何使用该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)，可以通过在一段时间内对系统执行文件级别的检查来全面地验证系统。

以下是本章中信息的列表：

- 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 105 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级别运行的文件跟踪工具。使用 BART，您可以迅速、轻松、可靠地收集有关安装在已部署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线确定系统上已发生的文件级别更改。可以使用 BART 从完全安装并配置的系统创建基准线或控制清单。之后，可以将此基准线与系统快照进行比较，以生成一个报告，列出自从该系统安装以来发生的文件级别更改。

`bart` 命令是一个标准 UNIX 命令。可以将 `bart` 命令的输出重定向到文件供以后处理。

BART 功能

BART 在设计时关注了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个时间段内的清单。然后，需要验证系统的文件时，可以通过比较新旧清单来生成报告。另一种使用 BART 的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 的其他优点和用途包括：

- 可用于高效且简便地为运行 Oracle Solaris 软件的系统创建文件级别的目录。
- 可用于定义要监视的文件，还可在必要时修改配置文件。借助这种灵活性，您可以监视本地定制设置，并可轻松高效地重新配置软件。
- 确保系统运行的是可靠的软件。
- 允许您监视系统在文件级别随时间发生的更改，从而帮助找到损坏或异常的文件。
- 帮助您解决系统性能问题。

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

可以使用 `bart create` 命令创建系统在特定时间的文件级别快照。输出是包含文件和文件属性信息的目录，称为**清单**。清单列出了有关系统上所有文件或特定文件的信息。它包含有关文件属性的信息，其中可能包括一些唯一标识信息，如 MD5 校验和。有关 MD5 校验和的更多信息，请参见 [md5\(3EXT\)](#) 手册页。可以存储清单，也可在客户机与服务器系统间传输清单。

注 - BART 不会跨越文件系统边界，除非文件系统属于同一类型。此约束可提高 `bart create` 命令输出的可预测性。例如，在不带参数的情况下，`bart create` 命令会创建根 (/) 目录下所有 ZFS 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 创建目录。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 105 页中的“BART 清单文件格式”。

BART 报告

该报告工具有三个输入：两个要比较的清单和一个由用户提供的可选 `rules` 文件（用于指明要标记的差异）。

可以使用 `bart compare` 命令比较两个清单：**控制清单**和**测试清单**。准备这些清单时使用的文件系统、选项和 `rules` 文件必须与用于 `bart create` 命令的相同。

`bart compare` 命令的输出是一份报告，其中基于每个文件列出了两个清单间的差异。**差异**是指上述两个清单中所列出的某个特定文件的任何属性的变化。两个清单间文件项的添加或删除也被视为差异。

报告差异时有两个控制级别：

- 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个清单间的差异所需的开销高。创建清单之后，即可通过运行以不同 **rules** 文件作为输入的 **bart compare** 命令，从不同方面比较这些清单。

有关 BART 报告的更多信息，请参见第 107 页中的“BART 报告”。

BART Rules 文件

Rules 文件是用作 **bart** 命令的可选输入的文本文件。此文件使用包含和排除规则。**Rules** 文件用于创建定制清单和报告。使用 **rules** 文件，可以用简洁的语法指定要在目录中列出的文件集，以及要监视任何给定文件集的哪些属性。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。使用 **rules** 文件，可以执行以下任务：

- 使用 **bart create** 命令创建一个清单，其中列出有关系统上所有文件或特定文件的信息。
- 使用 **bart compare** 命令生成一个报告，用于监视文件系统的特定属性。

注 – 可以创建多个 **rules** 文件，用于不同目的。但是，如果使用某个 **rules** 文件创建了清单，则比较清单时必须仍使用该 **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信息。如果 **rules** 文件确实包含错误信息，则也会报告这些用户错误。

要使用 **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。在创建 **rules** 文件之前，请首先确定您要监视系统上的哪些文件和文件属性。根据要完成的目标，可以使用 **rules** 文件来创建清单、比较清单或用于其他目的。

有关 BART **rules** 文件的更多信息，请参见第 106 页中的“BART Rules 文件格式”和 [bart_rules\(4\)](#) 手册页。

使用 BART（任务）

您可以使用一般用户、超级用户或承担角色的用户身份来运行 `bart` 命令。如果以一般用户身份运行 `bart` 命令，则只能在目录中列出和监视您有权访问的文件（如起始目录中的文件）。以超级用户身份运行 `bart` 命令的优势是，您创建的清单中会包含有关要监视的隐藏文件和私有文件的信息。如果需要在目录中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，请以超级用户身份运行 `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 127 页中的“基于角色的访问控制（概述）”。

BART 安全注意事项

以超级用户身份运行 `bart` 命令会使任何人都能读取其输出。此输出可能包含私有的文件名。如果以超级用户身份运行 `bart` 命令，请采取适当的措施保护输出。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。除非另有说明，否则以超级用户身份运行 `bart` 命令为可选操作。

使用 BART（任务列表）

任务	说明	参考
创建 BART 清单。	生成有关系统上安装的每个文件的信息列表。	第 97 页中的“如何创建清单”
创建定制 BART 清单。	生成有关系统上安装的特定文件的信息列表。	第 98 页中的“如何定制清单”
比较 BART 清单。	生成比较一个系统在一段时间内发生的更改的报告。 或者，生成将一个或几个系统与控制系统比较的报告。	第 99 页中的“如何比较同一个系统在一段时间内的清单” 第 101 页中的“如何比较不同系统的清单”
（可选的）定制 BART 报告。	通过以下方法之一生成定制 BART 报告： ■ 指定属性。 ■ 使用 <code>rules</code> 文件。	第 103 页中的“如何通过指定文件属性定制 BART 报告” 第 104 页中的“如何通过使用 Rules 文件定制 BART 报告”

▼ 如何创建清单

可以在初始安装 Oracle Solaris 软件之后立即创建系统的清单。此类型清单可提供用于比较同一个系统在一段时间内发生的更改的基准线。或者，也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创建快照，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线配置的同步。

开始之前 要创建系统清单，必须采用 root 角色。

1 在安装了 Oracle 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。由规则指定的所有路径都会被解释为此目录的相对路径。清单中报告的所有路径均为此目录的相对路径。
- I 接受要在目录中列出的单个文件的列表（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）。
- 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。

2 检查清单的内容。

3 保存清单供将来使用。

为清单选择一个有意义的名称。例如，使用创建清单的系统名称及日期。

示例 6-1 创建可列出有关系统上每个文件的信息的清单

如果运行 `bart create` 命令时不带任何选项，则会在目录中列出有关系统上安装的每个文件的信息。基于一个中心映像安装多个系统时，可使用此类清单作为基准线。或者，在需要确保安装完全一致时使用此类清单运行比较。

例如：

```
# bart create
! Version 1.1
! HASH SHA256
! Wednesday, September 07, 2011 (22:22:27)
# 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
3ebc418eb5be3729ffe7e54053be2d33ee884205502c81ae9689cd8cca5b0090 0 0
.
.
.
/zone D 512 40755 user::rwx group::r-x,mask:r-x,other:r-x 3f81e892
154de3e7bdf0d57a074c9fae0896a9e2e04bebfe5e872d273b063319e57f334 0 0
.
.
.
```

每个清单都包括一个文件头和多个文件项。每个清单文件项均单独占据一行，具体取决于文件类型。例如，对于以上输出中的每个清单项，类型 **F** 指定文件，而类型 **D** 指定目录。同时列出的还有关于大小、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顺序排列，以便正确地处理特殊字符。所有项均按文件名以升序排列。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）都会在排序之前将非标准字符用引号括起来。

以 **!** 开头的行提供有关清单的元数据。清单版本行指明清单规范版本。散列行指明所使用的散列机制。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。

▼ 如何定制清单

可以通过以下方法之一定制清单：

- 指定子树

为系统上的单个子树（而不是大目录的全部内容）创建清单是一种监视特定文件变化的高效方法。可以为系统上的特定子树创建基准线清单，然后定期创建该子树的测试清单。使用 **bart compare** 命令可将控制清单与测试清单进行比较。使用此选项，可以高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。

- 指定文件名

由于创建包含整个系统信息的清单耗时更长，占用空间更大，所需的开销也更高，因此在仅需要列出有关系统上一个或多个特定文件的信息时可选择使用 **bart** 命令的此选项。

- 使用 rules 文件

可以使用 **rules** 文件创建定制清单来列出有关给定系统上特定文件和特定子树的信息，还可以使用 **rules** 文件监视特定的文件属性。通过使用 **rules** 文件创建和比较清单，您可以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一个应用于创建的每个清单或生成的每个报告包含的所有文件的全局属性定义。

开始之前 您必须是 root 角色。

- 1 确定要在目录中列出和监视的文件。
- 2 安装了 Oracle Solaris 软件之后，使用以下选项之一创建定制清单：

- 指定子树：

```
# bart create -R root-directory
```

- 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
```

- 使用 rules 文件：

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

- 3 检查清单的内容。
- 4 保存清单供将来使用。

▼ 如何比较同一个系统在一段时间内的清单

要监视同一个系统在一段时间内的文件级别更改时，可使用此过程。此类型的清单可以帮助您找到损坏或异常的文件，检测安全违规，或对系统的性能问题进行故障排除。

开始之前 要创建并比较包含公共对象的清单，必须采用 root 角色。

- 1 在安装了 Oracle Solaris 软件之后，创建一个要在系统上监视的文件的控制清单。

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

- 2 在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
```

- 3 将控制清单与测试清单进行比较。

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

-r 用于此次比较的 rules 文件的名称。将 -r 选项和 - 一起使用表示将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 生成标准非本地化输出以进行程序解析的程序模式。
- control-manifest* 控制系统的 bart create 命令的输出。
- test-manifest* 测试系统的 bart create 命令的输出。

4 检查 BART 报告中的异常情况。

示例 6-2 比较同一个系统在一段时间内的清单

此示例说明如何监视 /etc 目录在两个时间点之间发生的更改。通过此类比较，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.090711
! Version 1.1
! HASH SHA256
! Wednesday, September 07, 2011 (11:11:17)
# 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
/.cpr_config F 2236 100644 owner@:read_data/write_data/append_data/read_xattr/write_xattr/read_attributes/write_attributes/read_acl/write_acl/write_owner/synchronize:allow,group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow,everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
4e271c59 0 0 3ebc418eb5be3729ffe7e54053be2d33ee884205502c81ae9689cd8cca5b0090
/.login F 1429 100644 owner@:read_data/write_data/append_data/read_xattr/write_xattr/read_attributes/write_attributes/read_acl/write_acl/write_owner/synchronize:allow,group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow,everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
4bf9d6d7 0 3 ff6251a473a53de68ce8b4036d0f569838cff107caf1dd9fd04701c48f09242e
.
.
.
```

- 在需要监视 /etc 目录的更改时创建测试清单。

```
# bart create -R /etc > system1.test.101011
Version 1.1
! HASH SHA256
! Monday, October 10, 2011 (10:10:17)
# 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
/.cpr_config F 2236 100644 owner@:read_data/write_data/append_data/read_xattr/wr
```



```
ite_xattr/read_attributes/write_attributes/read_acl/write_acl/write_owner/synchron
onize:allow,group@:read_data/read_xattr/read_attributes/read_acl/synchronize:all
ow,everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
4e271c59 0 0 3ebc418eb5be3729ffe7e54053be2d33ee884205502c81ae9689cd8cca5b0090
.
.
.
```

- 将控制清单与测试清单进行比较。

```
# bart compare system1.control.090711 system1.test.101011
/security/audit_class
mtime 4f272f59
```

上面的输出表明自控制清单创建以来，`audit_class` 文件的修改时间已发生变化。此报告可以用于检查所有权、日期、内容或任何其他文件属性是否发生了更改。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

可以运行系统间的比较，这样您能够迅速确定在基准线系统与其他系统之间是否存在任何文件级别差异。例如，如果已经在基准线系统上安装了特定版本的 Oracle 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，则可以创建这些系统的测试清单，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。

开始之前 要比较系统清单，必须采用 `root` 角色。

- 1 安装了 Oracle Solaris 软件之后，创建一个控制清单。

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

- 2 保存该控制清单。

- 3 在测试系统上，使用相同的 `bart` 选项创建一个清单，并将输出重定向到文件。

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

为测试清单选择一个独特而有意义的名称。

- 4 将测试清单保存在系统上的中心位置，直至可以开始比较清单为止。

- 5 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 NFS 挂载的系统，请使用 FTP 或其他可靠方法，将控制清单复制到测试系统。

- 6 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
```
- 7 检查 BART 报告中的异常情况。
- 8 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 6-3 将不同系统的清单与控制系统的清单进行比较

此示例说明如何通过将控制清单与不同系统的测试清单进行比较，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.090711
! Version 1.1
! HASH SHA256
! Wednesday, September 07, 2011 (11:11:17)
# 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
/2to3 F 105 100555 owner@:read_data/read_xattr/write_xattr/execute/read_attributes/write_attributes/read_acl/write_acl/write_owner/synchronize:allow,group@:read_data/read_xattr/execute/read_attributes/read_acl/synchronize:allow,everyone@:read_data/read_xattr/execute/read_attributes/read_acl/synchronize:allow 4bf9d261 0
2 154de3e7bdfd0d57a074c9fae0896a9e2e04bebf5e872d273b063319e57f334
/7z F 509220 100555 owner@:read_data/read_xattr/write_xattr/execute/read_attributes/write_attributes/read_acl/write_acl/write_owner/synchronize:allow,group@:read_data/read_xattr/execute/read_attributes/read_acl/synchronize:allow,everyone@:read_data/read_xattr/execute/read_attributes/read_acl/synchronize:allow 4dad48a 0
2 3ecd418eb5be3729ffe7e54053be2d33ee884205502c81ae9689cd8cca5b0090
...
```

- 为需要与控制系统进行比较的每个系统创建一个测试清单。

```
# bart create -R /usr/bin > system2-manifest.101011
! Version 1.1
! HASH SHA256
! Monday, October 10, 2011 (10:10:22)
# 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
/2to3 F 105 100555 owner@:read_data/read_xattr/write_xattr/execute/read_attributes/write_attributes/read_acl/write_acl/write_owner/synchronize:allow,group@:read_data/read_xattr/execute/read_attributes/read_acl/synchronize:allow,everyone@:read_data/read_xattr/execute/read_attributes/read_acl/synchronize:allow 4bf9d261 0
2 154de3e7bdfd0d57a074c9fae0896a9e2e04bebf5e872d273b063319e57f334
...
```

- 需要比较清单时，将清单复制到同一个位置。

```
# 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
```

上面的输出表明 `/usr/bin` 目录中 `su` 文件的组 ID 与控制系统中的相应组 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，或者是否可能有人篡改了文件。

▼ 如何通过指定文件属性定制 BART 报告

此过程为可选过程，说明如何通过从命令行指定文件属性来定制 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线清单，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时运行 `bart compare` 命令，并指定不同的属性。通过从命令行指定不同的文件属性，可以针对同一清单运行不同类型的比较。

开始之前 您必须是 root 角色。

- 1 确定需要监视的文件属性。
- 2 安装了 Oracle Solaris 软件之后，创建一个控制清单。
- 3 在需要监视更改时创建测试清单。
按照控制清单的准备方式准备测试清单。
- 4 比较这些清单。

例如：

```
# bart compare -i dirmtime,lnmtime,mtime control-manifest.121503 \
test-manifest.010504 > bart.report.010504
```

请注意，可使用逗号分隔在命令行语法中指定的各个属性。

- 5 检查 BART 报告中的异常情况。

▼ 如何通过使用 Rules 文件定制 BART 报告

此过程也是可选过程，说明如何通过使用 rules 文件作为 `bart compare` 命令的输入来定制 BART 报告。通过使用 rules 文件，可以定制 BART 报告，从而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 文件针对同一清单运行不同的比较。

开始之前 您必须是 root 角色。

- 1 确定需要监视的文件和文件属性。
- 2 使用文本编辑器创建包含适当指令的 rules 文件。
- 3 安装了 Oracle 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
```
- 4 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
```
- 5 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
```
- 6 检查 BART 报告中的异常情况。

示例 6-4 通过使用 Rules 文件定制 BART 报告

以下 rules 文件同时包含 `bart create` 命令和 `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，该 rules 文件还指示 `bart compare` 命令仅跟踪同一个目录的大小和内容的更改。

```
# 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
```

- 使用所创建的 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 命令报告了 /usr/bin 目录中的差异。该输出表明删除了 /usr/bin/ypcat 文件，添加了 /usr/bin/gunzip 文件。

BART 清单、Rules 文件和报告（参考）

本节介绍 BART 使用和创建的文件的格式。

BART 清单文件格式

每个清单文件项均单独占据一行，具体取决于文件类型。每个项都以 *fname*（即文件名）开头。为了避免因文件名中嵌入特殊字符所导致的解析问题，已对文件名进行了编码。有关更多信息，请参见第 106 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，其中包含 `acltotext()` 的输出。

uid 此项的所有者的数字用户 ID。

gid 此项的所有者的数字组 ID。

dirmtime 目录的上次修改时间，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。

<i>lnmtime</i>	链接的上次修改时间，以秒为单位，从 1970 年 1 月 1 日 00:00:00 UTC 开始计算。
<i>mtime</i>	文件的上次修改时间，以秒为单位，从 1970 年 1 月 1 日 00:00:00 UTC 开始计算。
<i>contents</i>	文件的校验和值。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 -。
<i>dest</i>	符号链接的目标。
<i>devnode</i>	设备节点的值。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。该工具将忽略以 **#** 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 6-5 Rules 文件格式

```
<Global CHECK/IGNORE Directives>
<subtree1> [pattern1..]
<IGNORE/CHECK Directives for subtree1>

<subtree2> [pattern2..]
<subtree3> [pattern3..]
<subtree4> [pattern4..]
<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。

属性关键字如下：

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

`all` 关键字表示所有文件属性。

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，以使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，如包含嵌入回车的文件名。使用 `rules` 规范语言可表述复杂的文件名过滤条件；如果仅使用 `shell` 语法会很难表述这些条件，而且效率也不高。

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所有文件，但已修改的目录时间戳(`dirmtime`)除外，如以下示例中所示：

```
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。

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

此报告机制提供了两种类型的输出：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` 命令对两个系统清单进行比较，则会生成一个文件差异列表。

例如：

```
filename attribute control:xxxx test:yyyy
```

filename 在控制清单与测试清单中不同的文件的名称。

attribute 进行比较的清单之间存在差异的文件属性的名称。*xxxx* 是控制清单中的属性值，*yyyy* 是测试清单中的属性值。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。输出表明 `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` 命令时使用了 `-p` 选项，就会生成程序输出。此输出以适合程序操作的格式生成。程序输出可以轻松被其他程序解析，目的在于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与测试清单之间存在差异的文件属性的说明

有关 `bart` 命令支持的属性的列表，请参见第 107 页中的“Rules 文件属性”。

有关 BART 的更多信息，请参见 [bart\(1M\)](#) 手册页。

控制对文件的访问（任务）

本章介绍了如何保护 Oracle Solaris 中的文件，还介绍了如何保护系统免受其权限可能会危及系统安全的文件的影响。

注 – 要使用访问控制列表 (access control list, ACL) 保护 ZFS 文件，请参见 [《Oracle Solaris 管理：ZFS 文件系统》](#) 中的第 8 章“使用 ACL 和属性保护 Oracle Solaris ZFS 文件”。

以下是本章中信息的列表。

- 第 109 页中的“使用 UNIX 权限保护文件”
- 第 116 页中的“防止可执行文件危及安全”
- 第 117 页中的“使用 UNIX 权限保护文件（任务列表）”
- 第 122 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

下表介绍了用于监视以及保证文件和目录安全的命令。

表 7-1 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	更改文件的所有权。	chown(1)

表 7-1 保证文件和目录安全的命令 (续)

命令	说明	手册页
chgrp	更改文件的组所有权。	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)

文件和目录的所有权

传统的 UNIX 文件权限可以为三类用户指定所有权：

- **用户**—文件或目录的所有者，通常为创建该文件的用户。文件的所有者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**—所有其他不是文件所有者和组成员的用户。

文件所有者通常可以指定或修改文件权限。此外，`root` 帐户可以更改文件的所有权。要覆盖系统策略，请参见[示例 7-2](#)。

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 7-2 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 7-2 文件和目录权限 (续)

符号	权限	对象	说明
w	写	目录	指定的用户可以列出目录中的文件。
		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 <code>exec(2)</code> 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所有者（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` 授予管理程序的所有权。要搜索并列出所有使用此特殊权限的文件，请参见第 123 页中的“[如何查找具有特殊文件权限的文件](#)”。

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所有者的访问权限。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。具有 `setuid` 权限的 `passwd` 命令类似如下：

```
-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予访问权限。`/usr/bin/mail` 命令具有 `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` 授予对此类程序的组访问权限。要搜索并列出所有使用此权限的文件，请参见第 123 页中的[“如何查找具有特殊文件权限的文件”](#)。

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，则只有文件所有者、目录所有者或特权用户才可以删除文件。`root` 用户是特权用户的一个示例。`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 7-5](#)。

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。文本文件具有 666 权限，该权限对所有用户授予读写权限。目录和可执行文件具有 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 shell 初始化文件（如 `.bashrc` 和 `.kshrc.user`）中的系统缺省值。管理员还可以设置 `/etc/profile` 文件中的缺省值。

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` 命令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。

下表显示了一些典型 `umask` 值及其对可执行文件的影响。

表 7-3 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。您必须是超级用户或是文件或目录的所有者，才能更改其权限。

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所有者、组和其他用户的权限。例如，值 `644` 为所有者设置读写权限，为组和其他用户设置只读权限。

表 7-4 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 7-5 在符号模式下设置文件权限

符号	功能	说明
u	<i>who</i>	用户（所有者）
g	<i>who</i>	组
o	<i>who</i>	其他用户
a	<i>who</i>	All（全部）
=	<i>operator</i>	赋值
+	<i>operator</i>	新增
-	<i>operator</i>	删除
r	<i>permissions</i>	读
w	<i>permissions</i>	写
x	<i>permissions</i>	执行
l	<i>permissions</i>	强制锁定，setgid 位打开，组执行位关闭
s	<i>permissions</i>	setuid 或 setgid 位打开
t	<i>permissions</i>	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 7-6 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

表 7-6 在绝对模式下设置特殊文件权限 (续)

八进制值	特殊文件权限
4	setuid

使用访问控制列表保护 UFS 文件

传统的 UNIX 文件保护可为以下三类用户提供读写和执行权限：文件所有者、文件组和其他用户。在 UFS 文件系统中，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：

- 为文件所有者、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

注 - 对于 ZFS 文件系统中的 ACL 以及 NFSv4 文件中的 ACL，请参见《Oracle Solaris 管理：ZFS 文件系统》中的第 8 章“使用 ACL 和属性保护 Oracle Solaris ZFS 文件”。

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。

在 UFS 文件系统中，通过 `setfacl` 命令在文件中设置 ACL 项。UFS ACL 项由以下字段（使用冒号进行分隔）组成：

- 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`（文件所有者）或 `mask`（ACL 掩码）。
- 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。

有关 UFS 文件系统 ACL 的更多信息，请参见 Oracle Solaris 10 发行版的《系统管理指南：安全性服务》。

防止可执行文件危及安全

程序读写栈中的数据。通常，程序从内存中专为代码指定的只读部分开始执行。一些可导致栈中的缓冲区溢出的攻击尝试向栈中插入新代码，从而致使程序执行该代码。删除栈内存的执行权限可使这些攻击无法成功。换句话说，大多数程序可以在不使用可执行栈的情况下正常工作。

64 位进程的栈通常不可执行。通过 `noexec_user_stack` 变量，您可以指定 32 位进程的栈是否可执行。为了符合 32 位 SPARC ABI，缺省值为零，即指定栈可执行。

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` 文件中设置此日志记录，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 `noexec_user_stack` 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` 文件中将日志变量 `noexec_user_stack_log` 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。也可以使用 `-M/usr/lib/ld/map.noexecstk` 编译您的程序，从而忽略系统级设置而使栈不可执行。

保护文件（任务）

以下过程使用 UNIX 权限保护文件、查找有安全风险的文件，并保护系统免受这些文件的危害。

使用 UNIX 权限保护文件（任务列表）

以下任务列表列出了用于列出文件权限、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 117 页中的“如何显示文件信息”
更改本地文件所有权。	第 118 页中的“如何更改文件的所有者” 第 119 页中的“如何更改文件的组所有者”
更改本地文件权限。	第 119 页中的“如何在符号模式下更改文件权限” 第 120 页中的“如何在绝对模式下更改文件权限” 第 121 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    显示包括用户所有权、组所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 7-1 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 4960
drwxr-xr-x  2 root    sys          64 Dec  8 11:57 ./
drwxr-xr-x 39 root    root         41 Dec  8 15:20 ../
-r-xr-xr-x  1 root    bin        21492 Dec  1 20:55 autopush*
-r-xr-xr-x  1 root    bin       33680 Oct  1 11:36 beadm*
-r-xr-xr-x  1 root    bin     184360 Dec  1 20:55 bootadm*
lrwxrwxrwx  1 root    root         21 Jun  7 2010 bpgetfile -> ...
-r-xr-xr-x  1 root    bin       86048 Dec  1 20:55 cryptoadm*
-r-xr-xr-x  1 root    bin       12828 Dec  1 20:55 devprop*
-r-xr-xr-x  1 root    bin     130132 Dec  1 20:55 dhcpagent*
-r-xr-xr-x  1 root    bin       13076 Dec  1 20:55 dhcpinfo*
```

每一行按以下顺序显示了有关文件的信息：

- 文件类型—例如，`d`。有关文件类型的列表，请参见第 110 页中的“文件和目录的所有权”。
- 权限—例如，`r-xr-xr-x`。有关说明，请参见第 110 页中的“文件和目录的所有权”。
- 硬链接数—例如，`2`。
- 文件的所有者—例如，`root`。
- 文件的组—例如，`bin`。
- 文件的大小（以字节为单位）—例如，`21308`。
- 文件创建日期或上次更改日期—例如，`Dec 9 15:55`。
- 文件名—例如，`dhcinfo`。

▼ 如何更改文件的所有者

开始之前 如果您不是文件或目录的所有者，则必须分配有 Object Access Management（对象访问管理）权限配置文件。要更改作为 [public object（公共对象）](#) 的文件，您必须是超级用户。

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 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

3 更改文件的所有者。

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

4 检验文件的所有者是否已更改。

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

已挂载 NFS 的文件系统对所有权和组的更改有更多限制。有关更多信息，请参见《Oracle Solaris 管理：网络服务》中的第 6 章“访问网络文件系统（参考）”。

示例 7-2 允许用户更改其自身文件的所有权

安全注意事项—您应该有合理理由将 `rstchown` 变量的设置更改为零。缺省设置阻止用户列出属于其他用户的文件以便规避空间配额。

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，文件所有者可以使用 `chown` 命令将文件的所有权更改为另一用户。通过此设置，文件所有者还可以使用 `chgrp` 命令将文件的组所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。

▼ 如何更改文件的组所有权

开始之前 如果您不是文件或目录的所有者，则必须分配有 Object Access Management（对象访问管理）权限配置文件。要更改作为 [public object（公共对象）](#) 的文件，您必须是超级用户。

- 1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

- 2 更改文件的组所有权。

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

有关设置组的信息，请参见《Oracle Solaris 管理：常见任务》中的第 2 章“管理用户帐户和组（概述）”。

- 3 检验文件的组所有权是否已更改。

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

另请参见[示例 7-2](#)。

▼ 如何在符号模式下更改文件权限

在以下过程中，用户更改其所拥有的文件的权限。

- 1 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。有关有效符号的列表，请参见[表 7-5](#)。

filename 指定文件或目录。

2 检验文件的权限是否已更改。

```
% ls -l filename
```

注 – 如果您不是文件或目录的所有者，则必须分配有 Object Access Management（对象访问管理）权限配置文件。要更改作为 **public object（公共对象）** 的文件，您必须是超级用户。

示例 7-3 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、组和其他用户的本地文件添加读和执行权限。

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

在以下示例中，组的读、写和执行权限被指定给本地文件。

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

▼ 如何在绝对模式下更改文件权限

在以下过程中，用户更改其所拥有的文件的权限。

1 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所有者、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 7-4](#)。

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。

2 检验文件的权限是否已更改。

```
% ls -l filename
```

注 – 如果您不是文件或目录的所有者，则必须分配有 Object Access Management（对象访问管理）权限配置文件。要更改作为 **public object（公共对象）** 的文件，您必须是超级用户。

示例 7-4 在绝对模式下更改权限

在以下示例中，将对公众公开的目录的权限从 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 jdoe  staff    6023 Aug  5 12:06 public_dir
# chmod 755 public_dir
# ls -ld public_dir
drwxr-xr-x 1 jdoe  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 jdoe  staff    6023 Aug  5 12:06 my_script
% chmod 700 my_script
% ls -l my_script
-rwx----- 1 jdoe  staff    6023 Aug  5 12:06 my_script
```

▼ 如何在绝对模式下更改特殊文件权限

开始之前 如果您不是文件或目录的所有者，则必须分配有 Object Access Management（对象访问管理）权限配置文件。要更改作为 **public object（公共对象）** 的文件，您必须是超级用户。

- 1 成为具有所需安全属性的管理员。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。

- 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 7-6。

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 7-5 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 jdoe    staff      512 May 15 15:27 public_dir
```

防止程序受到安全风险（任务列表）

以下任务列表列出了有关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
查找具有特殊权限的文件。	查找设置了 <code>setuid</code> 位，但不归 <code>root</code> 用户拥有的文件。	第 123 页中的“如何查找具有特殊文件权限的文件”
防止可执行栈溢出。	防止程序利用可执行栈。	第 124 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。	停止记录可执行栈消息。	示例 7-7

▼ 如何查找具有特殊文件权限的文件

此过程查找可能未经授权在程序中使用 `setuid` 和 `setgid` 权限的情况。可疑可执行文件为用户而不是 `root` 或 `bin` 授予所有权。

开始之前 您必须是 `root` 角色。

1 使用 `find` 命令查找具有 `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`（可以是根目录（`/`）、`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。

2 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 111 页中的“`setuid` 权限”。

示例 7-6 查找具有 `setuid` 权限的文件

以下示例的输出显示，名为 `rar` 的组中的用户创建了一份 `/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 `/var/tmp/ckprm` 目录移至归档文件，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 32 位可执行栈的安全风险的说明，请参见第 116 页中的“防止可执行文件危及安全”。

开始之前 您必须是 root 角色。

- 1 编辑 `/etc/system` 文件，添加以下行：

```
set noexec_user_stack=1
```

- 2 重新引导系统。

```
# reboot
```

示例 7-7 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
# reboot
```

另请参见 有关更多信息，请参阅以下链接：

- http://blogs.oracle.com/gbrunett/entry/solaris_non_executable_stack_overview
- http://blogs.oracle.com/gbrunett/entry/solaris_non_executable_stack_continued
- http://blogs.oracle.com/gbrunett/entry/solaris_non_executable_stack_concluded

第 3 部分

角色、权限配置文件和特权

本部分介绍了基于角色的访问控制 (role-based access control, RBAC) 和进程权限管理。RBAC 组件包括角色、权限配置文件和授权。进程权限管理通过特权实现。与通过超级用户管理系统相比，将特权与 RBAC 结合使用是一种更为安全的管理方法。

- 第 8 章，使用角色和特权（概述）
- 第 9 章，使用基于角色的访问控制（任务）
- 第 10 章，Oracle Solaris 中的安全属性（参考）

使用角色和特权（概述）

Oracle Solaris 的基于角色的访问控制 (role-based access control, RBAC) 功能和 Oracle Solaris 的特权功能向超级用户提供了一个更为安全的替代方法。本章概述了有关 RBAC 与特权的信息。

以下是本章中概述信息的列表。

- 第 127 页中的“基于角色的访问控制（概述）”
- 第 138 页中的“特权（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，用于控制用户对通常仅限于 root 角色的任务的访问。通过对进程和用户应用安全属性，RBAC 可以向多个管理员分派超级用户功能。进程权限管理通过**特权**实现。用户权限管理通过 RBAC 实现。

- 有关进程权限管理的介绍，请参见第 138 页中的“特权（概述）”。
- 有关 RBAC 任务的信息，请参见第 9 章，使用基于角色的访问控制（任务）。
- 有关参考信息，请参见第 10 章，Oracle Solaris 中的安全属性（参考）。

RBAC：超级用户模型的替代方法

在传统 UNIX 系统中，root 用户（也称为超级用户）具有最高权限。以 root 身份运行的程序（即 `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，更改审计迹，读取机密记录以及关闭整个网络。被劫持的 `setuid` 程序可以在系统上执行任何操作。

与超级用户模型（管理员对系统要么具有全部控制权要么毫无控制权）相比，基于角色的访问控制 (Role-based access control, RBAC) 是一种更为安全的方法。使用 RBAC，可以在更精细的级别上强制执行安全策略。RBAC 采用**最小特权**的安全原

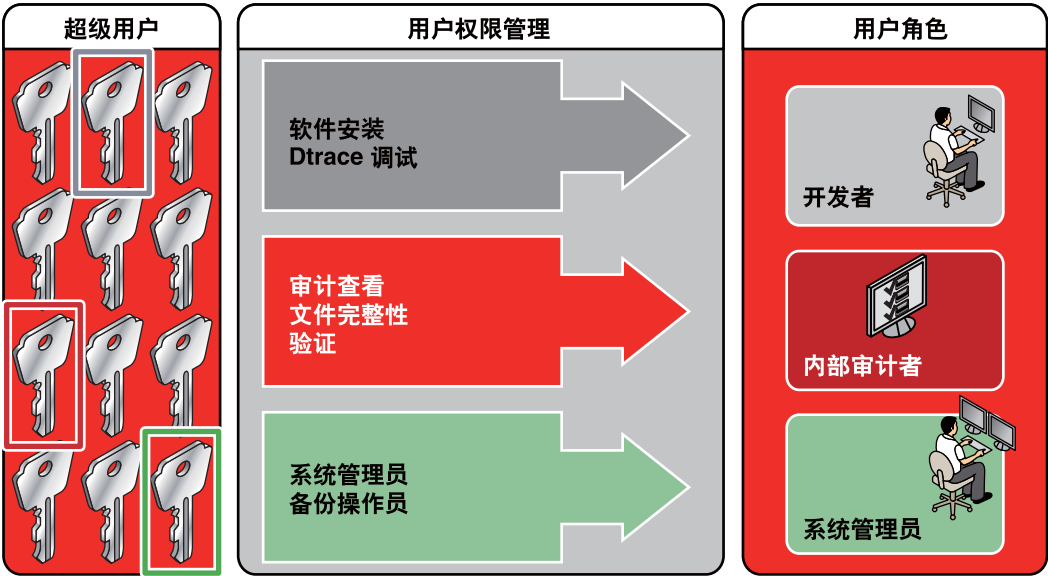
则。最小特权表示用户仅具有执行某项作业所必需的权限。一般用户具有足够特权来使用其应用程序、检查其作业状态、打印文件、创建新文件等。超出一般用户功能以外的功能将分组到各权限配置文件中。如果用户将要执行的作业要求具有某些超级用户功能才能执行，这些用户将承担拥有适当权限配置文件的角色。

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，用户可以承担某个角色，以执行要求具有某些超级用户功能才能执行的作业。预定义的权限配置文件随 Oracle Solaris 软件附带。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，System Administrator（系统管理员）权限配置文件允许帐户执行与安全无关的任务，如打印机管理和 cron 作业。也可将权限配置文件定义为提供限定功能。例如，Cron Management（计时程序管理）权限配置文件可管理 at 和 cron 作业。创建角色后，可以为这些角色指定广泛的功能或限定的功能，或者指定这两者。

下图说明了 RBAC 如何将权限分配给各可信方。

图 8-1 RBAC 的权限分配



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执行角色任务的可靠用户。这些用户使用自己的用户名进行登录。登录之后，用户将承担可以运行限定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，因此可实现各种安全策略。尽管 Oracle Solaris 附带的角色很少，但是可以轻松地配置多种角色。您可以将同一名称的权限配置文件作为大多数角色的基础：

- **root**—等效于 root 用户的功能强大的角色。但是，此 root 角色不能登录。一般用户必须先登录，再承担指定的 root 角色。缺省情况下将配置该角色。
- **System Administrator（系统管理员）**—可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。
- **Operator（操作员）**—可执行备份和打印机管理等操作的初级管理员角色。

注 – Media Backup（介质备份）权限配置文件提供对整个根文件系统的访问权限。因此，面向初级管理员设计 Media Backup（介质备份）和 Operator（操作员）权限配置文件时，必须确保用户可信。

可能还希望配置一个或多个安全角色。三个权限配置文件及其补充的配置文件处理安全性：信息安全、用户安全和区域安全。网络安全是信息安全权限配置文件中的补充配置文件。

并不一定要实现这些角色。角色是一种满足组织安全性需要的功能。一种策略是为安全、网络或防火墙管理等领域中的特殊用途管理员设置角色。另一种策略是创建一个功能强大的管理员角色以及一个高级用户角色。高级用户角色将用于那些获许修复其自己的系统部分的用户。

超级用户模型与 RBAC 模型可以共存。下表汇总了 RBAC 模型中可以存在的各个等级（从超级用户到限定一般用户）。该表包含可以在这两种模型中跟踪的管理操作。有关特权对系统的单独影响的概要信息，请参见表 8-2。

表 8-1 超级用户模型与具有特权的 RBAC 模型对照

系统上的用户功能	超级用户模型	RBAC 模型
是否可成为具有完全超级用户功能的超级用户	可以	可以
是否可使用具有完全用户功能的用户身份登录	可以	可以
是否可成为具有有限功能的超级用户	不可以	可以
是否可使用用户身份登录，有时具有超级用户功能	可以，仅限 <code>setuid</code> 程序	可以，使用 <code>setuid</code> 程序和 RBAC
是否可使用具有管理权能的用户身份登录，但是不具有完全超级用户功能	不可以	可以，使用 RBAC 以及直接指定的特权和授权

表 8-1 超级用户模型与具有特权的 RBAC 模型对照（续）

系统上的用户功能	超级用户模型	RBAC 模型
是否可使用功能少于一般用户的用户身份登录	不可以	可以，使用 RBAC 和已删除的特权
是否可跟踪超级用户操作	可以，通过审计 su 命令	可以，通过审计对 pfexec() 的调用 另外，审计迹中会出现已经承担 root 角色的用户的名称

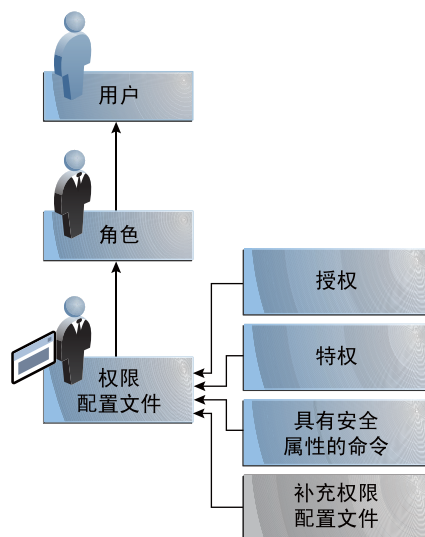
RBAC 元素和基本概念

Oracle Solaris 中的 RBAC 模型引入了以下元素：

- **授权**—一种权限，允许用户或角色执行某一类需要额外权限才能执行的操作。例如，安装过程中的安全策略会为一般用户提供 `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。
- **特权**—可以授予命令、用户、角色或系统的独立权限。特权可以保证进程成功执行。例如，`proc_exec` 特权允许进程调用 `execve()`。一般用户具有基本特权。要查看您的基本特权，请运行 `ppriv -vl basic` 命令。
- **安全属性**—允许进程执行某个操作的属性。在典型的 UNIX 环境中，安全属性允许进程执行原本禁止一般用户执行的操作。例如，`setuid` 和 `setgid` 程序具有安全属性。在 RBAC 模型中，授权和特权是除 `setuid` 和 `setgid` 程序之外的安全属性。可以将这些属性指定给某个用户。例如，具有 `solaris.device.allocate` 授权的用户可以分配设备供独占使用。特权可以置于某个进程上。例如，具有 `file_flag_set` 特权的进程可以设置不变的、未解除链接的或仅附加的文件属性。
- **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，需要有特权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 134 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安全属性的集合。一个权限配置文件可以包含授权、直接指定的特权、具有安全属性的命令以及其他权限配置文件。其他配置文件中的配置文件称为补充权限配置文件。权限配置文件提供了一种便捷的安全属性分组方法。
- **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（包括 root 角色）运行的系统中，超级用户是不必要的。超级用户功能会分配给不同的角色。例如，在有两种角色的系统中，将由其中的安全角色处理安全任务，而另一个角色负责处理与安全无关的系统管理任务。角色可以进行更细粒度的划分。例如，系统可以包括各种独立的管理角色，分别用于处理加密框架、打印机、系统时间、文件系统和审计。

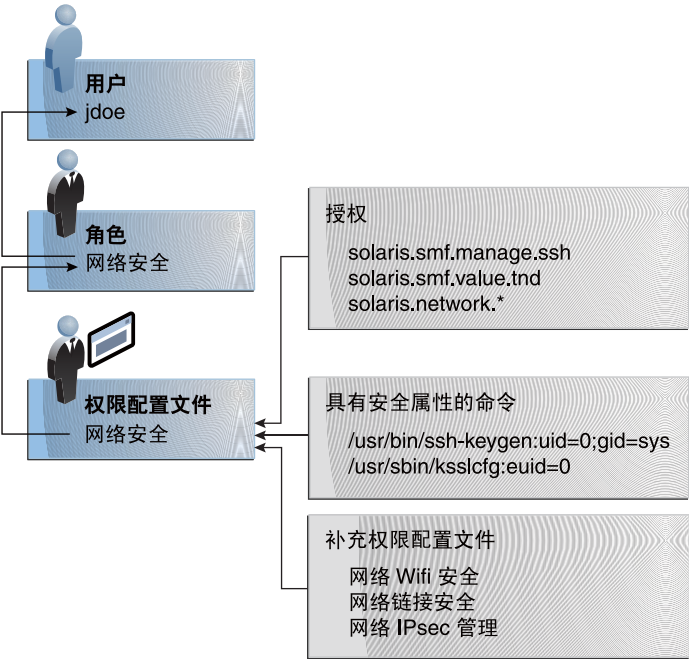
下图说明了各 RBAC 元素如何协同工作。

图 8-2 RBAC 元素关系



下图使用 Network Security（网络安全）角色和 Network Security（网络安全）权限配置文件说明了 RBAC 关系。

图 8-3 RBAC 元素关系示例



网络安全角色用于管理 IPsec、wifi 和网络链接。该角色指定给用户 jdoe。jdoe 可以通过切换到该角色然后提供角色口令来承担该角色。管理员可以定制角色以接受用户口令，而不是角色口令。

在图 8-3 中，Network Security（网络安全）权限配置文件指定给 Network Security（网络安全）角色。Network Security（网络安全）权限配置文件包含一些按顺序评估的补充配置文件：Network Wifi Security（网络 Wifi 安全）、Network Link Security（网络链接安全）和 Network IPsec Management（网络 IPsec 管理）。这些补充配置文件用于角色的主要任务。

Network Security（网络安全）权限配置文件有三个直接指定的授权，没有直接指定的特权，还有两个具有安全属性的命令。补充权限配置文件有直接指定的授权，其中两个包含具有安全属性的命令。在 Network Security（网络安全）角色中，jdoe 拥有这些配置文件中的所有指定授权，并可运行这些配置文件中所有具有安全属性的命令。jdoe 可以管理网络安全。

特权升级

Oracle Solaris 为管理员提供了安全性配置的极大灵活性。安装该软件后不允许进行 [privilege escalation](#)（[特权升级](#)）。当用户或进程获取的管理权限多于本应授予的权限时，才会发生特权升级。从这个意义上说，特权就是某种安全属性，而不仅是特权。

Oracle Solaris 软件仅包括指定给 root 角色的安全属性。在实施了其他安全保护措施的情况下，管理员可以将为 root 角色设计的属性指定给其他帐户，但必须谨慎进行此类指定。

以下权限配置文件和授权集可以升级非超级用户帐户的特权。

- **Media Restore（介质恢复）权限配置文件**—此配置文件存在，但不是任何其他权限配置文件的一部分。因为 Media Restore（介质恢复）配置文件提供对整个根文件系统的访问权限，因此使用它可能是一种特权升级。可以恢复故意更改的文件或替代介质。缺省情况下，root 角色包括此权限配置文件。
- **solaris.*.assign 授权**—这些授权存在，但未指定给任何权限配置文件或帐户。具有 solaris.*.assign 授权的帐户可以将安全属性指定给帐户本身未指定的其他帐户。例如，具有 solaris.profile.assign 授权的角色可以将权限配置文件指定给角色本身未指定的其他帐户。缺省情况下，只有 root 角色具有 solaris.*.assign 授权。

最佳做法是指定 solaris.*.delegate 授权，而不是 solaris.*.assign 授权。solaris.*.delegate 授权仅允许委托方将其拥有的安全属性指定给其他帐户。例如，指定有 solaris.profile.delegate 授权的角色可以将角色本身指定的权限配置文件指定给其他用户和角色。

有关会影响特权安全属性的升级，请参见第 195 页中的“防止特权升级”。

RBAC 授权

授权是指可以授予角色或用户的独立权限。授权在用户应用程序级别强制执行策略。

尽管可以将授权直接指定给角色或用户，但最佳做法是将授权归到权限配置文件中。然后，将该权限配置文件添加到某个角色中，再将该角色指定给用户。有关示例，请参见图 8-3。

包含单词 **delegate** 或 **assign** 的授权允许用户或角色将安全属性指定给其他帐户。

要阻止升级特权，请不要将 **assign** 授权指定给某个帐户。

- **delegate** 授权仅允许委托方将其拥有的安全属性指定给其他帐户。例如，分配有 solaris.profile.delegate 授权的角色可以将角色本身指定的权限配置文件指定给其他帐户。
- **assign** 授权允许指定者将帐户没有的安全属性指定给其他帐户。例如，具有 solaris.profile.assign 授权的角色可以将任何权限配置文件指定给其他帐户。

solaris.*.assign 授权已提供，但不包含在任何配置文件中。缺省情况下，只有 root 角色具有 solaris.*.assign 授权。

遵循 RBAC 的应用程序可以先检查用户的授权，然后再授予其访问该应用程序或应用程序中特定操作的权限。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 188 页中的“Authorizations（授权）”
- 第 190 页中的“auth_attr 数据库”
- 第 192 页中的“要求授权的选定命令”

授权和特权

特权在内核中强制执行安全策略。授权与特权之间的差别与强制执行安全策略的级别有关。如果没有适当的特权，内核可能会阻止进程执行特权操作。如果没有适当的授权，用户可能无法使用特权应用程序或在特权应用程序内执行与安全相关的操作。有关特权的更全面的介绍，请参见第 138 页中的“特权（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。使用安全属性（如 `UID=0`）、特权和授权，可以将应用程序成为特权应用程序。

检查 UID 和 GID 的应用程序

检查 `root` (`UID=0`) 或其他某个特殊 UID 或 GID 的特权应用程序在 UNIX 环境中已经存在很久。通过权限配置文件机制，可以隔离需要特定 ID 的命令。您不必更改任何人都可访问的命令的 ID，而是可以将具有指定安全属性的命令放置在某个权限配置文件中。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，该 ID 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，`reboot` 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，则需要将此实际 ID 指定给该命令。

检查特权的应用程序

特权应用程序可以检查特权的使用。通过 RBAC 权限配置文件机制，可以为需要安全属性的特定命令指定特权。然后，可以在权限配置文件中隔离具有指定安全属性的命令。这样，拥有此权限配置文件的用户或角色便可仅使用成功执行命令所需的特权来运行此命令。

检查特权的命令包括：

- Kerberos 命令，如 `kadmin`、`kprop` 和 `kdb5_util`
- 网络命令，如 `ipadm`、`routeadm` 和 `snoop`

- 文件和文件系统命令，如 `chmod`、`chgrp` 和 `mount`
- 控制进程的命令，如 `kill`、`pcrred` 和 `rcapadm`

要将具有特权的命令添加到权限配置文件中，请参见第 159 页中的“[如何创建或更改权限配置文件](#)”和 [profiles\(1\)](#) 手册页。要确定哪些命令可检查特定配置文件中的特权，请参见第 146 页中的“[如何查看所有已定义的安全属性](#)”。

检查授权的应用程序

Oracle Solaris 额外提供了用于检查授权的命令。根据定义，`root` 用户拥有所有授权。因此，`root` 用户可以运行任何应用程序。检查授权的应用程序包括：

- 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`。

要测试用于授权的脚本或程序，请参见[示例 9–16](#)。要编写需要授权的程序，请参见《[Oracle Solaris 11 开发者安全性指南](#)》中的“[关于授权](#)”。

RBAC 权限配置文件

权限配置文件是安全属性的集合，可以指定给角色或用户以执行需要管理权限的任务。一个权限配置文件可以包含授权、特权、具有指定安全属性的命令以及其他权限配置文件。在权限配置文件中指定的特权对所有命令都生效。权限配置文件还包含减少或扩展初始可继承权限集的条目和减少特权的限制集的条目。

有关权限配置文件的更多信息，请参见以下各节：

- [第 185 页中的“权限配置文件”](#)
- [第 190 页中的“prof_attr 数据库”](#)
- [第 191 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。角色的创建方式与用户帐户的常规创建方式相同。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权能。角色不能从其他角色或其他用户继承功能。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，此角色的属性将取代所有用户属性。角色信息存储在 `passwd`、`shadow` 和 `user_attr` 数据库中。可以审计角色的操作。有关设置角色的详细信息，请参见以下各节：

- [第 153 页中的“如何规划 RBAC 实现”](#)
- [第 155 页中的“如何创建角色”](#)
- [第 167 页中的“如何更改角色的安全属性”](#)

可以将一种角色指定给多个用户。所有可以承担相同角色的用户都具有同一个角色起始目录，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。缺省情况下，用户通过提供角色的口令来验证角色。管理员可以将系统配置为允许用户通过提供用户的口令来验证。有关过程，请参见 [第 172 页中的“如何让用户使用自己的口令承担角色”](#)。

角色不能直接登录。用户需要首先登录，然后承担角色。用户承担一种角色之后，如果不先退出当前角色，将无法承担其他角色。用户退出当前角色之后，才能承担其他角色。

`root` 是 Oracle Solaris 中的一个角色的事实阻止了匿名 `root` 登录。如果审计配置文件 `shell` 命令 `pfexec`，则审计迹会包含登录用户的实际 UID、用户承担的角色以及该角色执行的操作。要针对角色操作来审计系统或特定用户，请参见 [第 159 页中的“如何审计角色”](#)。

随软件提供的权限配置文件与角色相对应。例如，`System Administrator`（系统管理员）权限配置文件可用于创建 `System Administrator`（系统管理员）角色。要配置角色，请参见 [第 155 页中的“如何创建角色”](#)。

配置文件 Shell 和 RBAC

用户和角色可以从 [profile shell（配置文件 shell）](#) 运行特权应用程序。配置文件 `shell` 是一种特殊的 shell，可以识别权限配置文件中包含的安全属性。管理员可以将配置文件 `shell` 指定给特定用户以作为登录 shell，或者在该用户运行 `su` 命令时启动配置文件 `shell` 以承担某个角色。在 Oracle Solaris 中，每个 shell 均具有对应的配置文件 `shell`。例如，与 Bourne shell (`sh`)、bash shell (`csh`) 和 Korn shell (`ksh`) 相对应的配置文件 `shell` 分别是 `pfsh`、`pfbash` 和 `pfksh shell`。有关配置文件 `shell` 的列表，请参见 [pfexec\(1\)](#) 手册页。

已直接分配有权限配置文件及其登录 shell 不是配置文件 `shell` 的用户必须调用配置文件 `shell` 来运行具有安全属性的命令。有关可使用性和安全性的注意事项，请参见 [第 137 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 159 页中的“如何审计角色”](#)。

名称服务范围 and RBAC

名称服务范围是了解 RBAC 的一个重要概念。角色的范围可以限定为单个主机。或者，此范围也可以包括由某种名称服务（如 LDAP）提供服务的所有主机。在名称转换服务 `svc:/system/name-service/switch` 中指定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，则只会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权限。授权、特权和特权命令会分组到权限配置文件中。该权限配置文件会包括到某个角色中，再将该角色指定给用户。

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、特权和授权直接指定给用户。
- 可以将特权和授权直接指定给用户和角色。

但是，直接指定特权的做法并不安全。具有直接指定的特权的用户和角色可能会在每次内核需要该特权时覆盖安全策略。更安全的做法是在权限配置文件中将特权指定为某个命令的安全属性。这样，该特权只可由拥有此权限配置文件的用户用于该命令。

由于授权在用户级别执行，因此，与直接指定特权相比，直接指定授权的危险性会更小。但是，用户可以使用授权来执行高度安全的任务，如指定审计标志。

直接指定安全属性时的可用性注意事项

直接指定权限配置文件和安全属性可能会影响可用性：

- 直接指定的特权和授权，以及直接指定的权限配置文件中的命令和授权必须由配置文件 `shell` 解释才能生效。缺省情况下，未将配置文件 `shell` 指定给用户。
用户必须记得打开某个配置文件 `shell`，然后在该 `shell` 中执行命令。
- 单独指定授权是不可伸缩的。而且，直接指定的授权可能不足以执行某个任务。该任务可能需要特权命令。
权限配置文件用于将授权和特权命令捆绑在一起。它们也是可伸缩的。

特权（概述）

进程权限管理允许在命令、用户、角色和系统四个级别上对进程加以限制。Oracle Solaris 通过**特权**实现进程权限管理。特权可以降低与（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。特权与 RBAC 相结合，是传统超级用户模型的强大的替代模型。

- 有关 RBAC 的信息，请参见第 127 页中的“基于角色的访问控制（概述）”。
- 有关如何管理特权的信息，请参见第 175 页中的“使用特权（任务）”。
- 有关特权的参考信息，请参见第 194 页中的“特权”。

特权保护内核进程

特权是进程执行某项操作所需的独立权限。权限是在内核中实施的。如果程序在**基本特权集**界限内运行，则该程序也会在系统安全策略界限内运行。例如，`setuid` 程序就是在系统安全策略界限外部运行的程序。通过使用特权，程序可以不必调用 `setuid`。

特权以离散方式枚举可在系统上执行的操作种类。程序可以使用可使其成功执行的确切特权运行。例如，处理文件的程序可能需要 `file_dac_write` 和 `file_flag_set` 特权。有了此功能，就不需要以 `root` 身份运行该程序了。

以前，系统并未遵循特权模型，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。用户进程只能对用户的目录和文件执行操作。`root` 进程可以在系统上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`（即以 `root` 身份）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` 拥有的设备只能由该 `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` 进程获许执行的操作数比该进程完成其运行所需的操作数多。`setuid` 程序可能会受到来自随后将以无所不能的 `root` 用户身份运行的侵入者的威胁。同样，能够访问 `root` 口令的任何用户都可能会危及整个系统的安全。

与之相反，通过特权实施策略的系统允许在用户功能和 `root` 功能之间划分等级。可以为某个用户授予特权，允许其执行超出一般用户功能的活动，并可对 `root` 加以限制，使 `root` 拥有的特权比当前拥有的特权少。借助 RBAC，可以将使用特权运行的命令单独放置在某个权限配置文件中，并将其指定给一个用户或角色。表 8-1 汇总了 RBAC 与特权模型提供的用户功能和超级用户功能之间的等级。

特权模型比超级用户模型具有更好的安全性。已从进程中删除的特权不会被利用。进程特权防止程序或管理帐户获取对所有功能的访问权限。进程特权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，特权可以限制程序和进程，使其仅具备程序所需的功能。此功能称为**最小特权原则**。在实现最小特权的系统上，捕获某个进程的入侵者只能访问该进程所具有的那些特权，不会破坏其余部分的系统安全。

特权说明

可以根据特权范围对特权进行逻辑分组。

- **FILE 特权**—以字符串 `file` 开头的特权用于文件系统对象。例如，`file_dac_write` 特权允许在写入文件时覆盖自主访问控制。
- **IPC 特权**—以字符串 `ipc` 开头的特权允许覆盖 IPC 对象访问控制。例如，`ipc_dac_read` 特权允许进程读取受 DAC 保护的远程共享内存。
- **NET 特权**—以字符串 `net` 开头的特权授予对特定网络功能的访问权限。例如，`net_rawaccess` 特权允许设备连接到网络。
- **PROC 特权**—以字符串 `proc` 开头的特权允许进程修改进程本身的限定属性。**PROC** 特权包括影响非常有限的特权。例如，`proc_clock_highres` 特权允许进程使用高分辨率的计时器。
- **SYS 特权**—以字符串 `sys` 开头的特权为进程授予对各种系统属性的无限制访问权限。例如，`sys_linkdir` 特权允许进程建立和断开指向目录的硬链接。

其他逻辑组包括 `CONTRACT`、`CPC`、`DTRACE`、`GRAPHICS`、`VIRT`、`WIN` 和 `XVM`。

有些特权对系统的影响有限，而有些特权则具有广泛的影响。`proc_taskid` 特权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` 特权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\(5\)](#) 手册页提供了每种特权的说明。`ppriv -lv` 命令会将每种特权的说明显示在标准输出中。

具有特权与不具有特权的系统之间的管理差别

具有特权的系统与不具有特权的系统之间存在多处明显差别。下表列出了其中的一些差别。

表 8-2 具有特权的系统与不具有特权的系统之间的明显差别

功能	不具有特权	特权
守护进程	以 <code>root</code> 身份运行守护进程。	以 <code>daemon</code> 用户身份运行守护进程。 例如，以下守护进程已指定有相应的特权，并以 <code>daemon</code> 身份运行： <code>lockd</code> 、 <code>nfsd</code> 和 <code>rpcbind</code> 。

表 8-2 具有特权的系统与不具有特权的系统之间的明显差别 （续）

功能	不具有特权	特权
日志文件所有权	日志文件由 root 拥有。	现在，日志文件由创建此日志文件的 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。	错误消息反映特权的使用。 例如，chroot 失败的等效错误消息为 chroot: exec failed。
setuid 程序	程序使用 setuid 来完成不允许一般用户执行的任务。	许多 setuid 程序都已更改为使用特权运行。 例如，以下命令使用特权：audit、ikeadm、ipadm、ipsecconf、ping、traceroute 和 newtask。
文件权限	设备权限由 DAC 控制。例如，sys 组的成员可以打开 /dev/ip。	文件权限 (DAC) 不会预测谁可以打开设备。设备通过 DAC 和设备策略进行保护。 例如，/dev/ip 文件具有 666 权限，但是该设备只能由具有适当特权的进程打开。原始插槽仍受 DAC 保护。
审计事件	审计 su 命令的使用会涉及许多管理功能。	审计特权的使用会涉及大多数管理功能。pm、ps、ex、ua 和 as 审计类包括那些对设备策略和特权使用进行监视的审计事件。
进程	进程受进程的拥有者保护。	进程受特权保护。进程特权和进程标志显示为 /proc/<pid> 目录中的一个新项 priv。
调试	不引用核心转储中的任何特权。	核心转储的 ELF 注释部分在 NT_PRPRIV 和 NT_PRPRIVINFO 注释中包含有关进程特权和标志的信息。 ppriv 命令和其他命令显示了大小合适的集的正确数目。这些命令会将位集中的位正确映射到特权名称。

特权和系统资源

在 Oracle Solaris 发行版中，project.max-locked-memory 和 zone.max-locked-memory 资源控制可用于限制分配有 PRIV_PROC_LOCK_MEMORY 特权的进程的内存消耗。此特权允许进程锁定物理内存中的页。

如果将 PRIV_PROC_LOCK_MEMORY 特权指定给某个权限配置文件，则可为具有此特权的进程提供锁定所有内存的功能。作为一项保护措施，请设置资源控制以防止具有该特权的用户锁定所有内存。对于在非全局区域中运行的特权进程，请设置 zone.max-locked-memory 资源控制。对于在系统上运行的特权进程，请创建一个项目，然后设置 project.max-locked-memory 资源控制。有关这些资源控制的信息，请参见《Oracle Solaris 管理：Oracle Solaris Zones、Oracle Solaris 10 Zones 和资源管理》中的第 6 章“资源控制（概述）”和《Oracle Solaris 管理：Oracle Solaris Zones、Oracle Solaris 10 Zones 和资源管理》中的第 16 章“非全局区域配置（概述）”。

如何实现特权

每个进程都有四个特权集，用于确定进程是否可以使用某个特定的特权。内核会自动计算**有效特权集**。可以修改初始**可继承特权集**。通过编码来使用特权的程序可以减小该程序的**允许特权集**。可以缩小**限制特权集**。

- **有效特权集 (E)**—当前生效的特权集。进程可以将允许特权集中的特权添加到有效特权集，也可以从 E 中删除特权。
- **允许特权集 (P)**—可供使用的特权集。特权可通过继承或指定供程序使用。执行配置文件便是一种将特权指定给程序的方法。`setuid` 命令可将 `root` 具有的所有特权指定给程序。可从允许特权集中删除特权，但不能向该集中添加特权。已从 P 中删除的特权会自动从 E 中删除。

可识别特权的程序会从该程序的允许特权集中删除程序从不使用的特权。通过这种方法，程序或恶意进程将无法利用不必要的特权。有关可识别特权的程序的更多信息，请参见《[Oracle Solaris 11 开发者安全性指南](#)》中的第 2 章“开发特权应用程序”。

- **可继承特权集 (I)**—进程可通过调用 `exec` 继承的特权集。调用 `exec` 之后，允许特权集和有效特权集会相同，但 `setuid` 程序这种特殊情况除外。

对于 `setuid` 程序，调用 `exec` 之后，可继承特权集首先会受限制特权集的限制。然后，系统会从继承特权集 (I) 中删除限制特权集 (L) 中的所有特权，并将结果指定给此进程的 P 和 E。

- **限制特权集 (L)**—对可用于某个进程及其子进程的特权的外部限制。缺省情况下，限制特权集为所有特权。进程可以缩小限制特权集，但是永远不能扩展限制特权集。L 用来限制 I。因此，在执行 `exec` 命令时，L 限制 P 和 E。

如果为用户指定的配置文件中包含指定有特权的程序，则此用户通常可以运行该程序。在未修改的系统上，为此程序指定的特权在用户的限制特权集中。已指定给程序的特权会成为用户的允许特权集的一部分。要运行已分配有特权的程序，用户必须从配置文件 `shell` 运行此程序。

内核可以识别**基本特权集**。在未修改的系统上，每个用户的初始可继承特权集等同于登录时获取的基本特权集。虽然您不能修改基本特权集，但可以修改用户从基本特权集中继承的特权。

在未修改的系统上，用户登录时的特权集将类似如下：

```
E (Effective): basic
I (Inheritable): basic
P (Permitted): basic
L (Limit): all
```

因此，在登录时，所有用户在其可继承特权集、允许特权集和有效特权集中包含了基本特权集。用户的限制特权集等同于区域（全局区域或非全局区域）的缺省特权集。要在用户的有效特权集中加入更多特权，必须为该用户指定一个权限配置文件

件。此权限配置文件将包含已添加了特权的命令。还可以直接将特权指定给用户或角色，但这种特权指定可能会有风险。有关风险的介绍，请参见第 137 页中的“直接指定安全属性时的安全注意事项”。

进程如何获取特权

进程可以继承特权。或者，也可以为进程指定特权。进程从其父进程继承特权。登录时，用户的初始可继承特权集确定该用户的进程可用的特权。用户初始登录的所有子进程都从此特权集继承。

也可以直接将特权指定给程序、用户和角色。当某个程序需要特权时，可以在权限配置文件中将特权指定给该程序的可执行文件。系统会向获许运行程序的用户或角色指定包含该程序的配置文件。登录或进入配置文件 shell 时，如果在配置文件 shell 中键入了程序的可执行文件，则可使用特权运行该程序。例如，拥有 Object Access Management（对象访问管理）配置文件的角色可以使用 `file_chown` 特权运行 `chmod` 命令。

当某个角色或用户运行已直接分配有附加特权的程序时，该指定特权会添加到此角色或用户的可继承特权集中。指定有特权的程序的子进程会继承父进程的特权。如果子进程需要的特权比父进程的特权多，则必须直接为子进程指定这些额外特权。

privilege-aware（特权识别）通过编码使用特权的程序称为可识别特权的程序。可识别特权的程序可在程序执行过程中启用和禁用特权。要在生产环境中成功执行程序，必须为程序指定其可启用和禁用的特权。

有关可识别特权的代码示例，请参见《Oracle Solaris 11 开发者安全性指南》中的第 2 章“开发特权应用程序”。要为需要特权的程序指定特权，请参见示例 9-14。

指定特权

您作为安全管理员应负责指定特权。最佳做法是在权限配置文件中将特权指定给命令。然后，将该权限配置文件指定给角色或用户。

特权还可以直接指定给用户、角色或权限配置文件。如果您信任某些用户能够在其整个会话中会负责地使用某种特权，则可以直接指定该特权。适合直接指定的特权是具有有限影响的特权，如 `proc_clock_highres`。不适合直接指定的特权是具有广泛影响的特权，如 `file_dac_write`。

还可以拒绝用户或系统的特权。从用户或系统的初始可继承特权集或限制特权集中删除特权时必须小心谨慎。

扩展用户或角色的特权

用户和角色具有可继承特权集。不能扩展限制特权集，因为限制特权集最初包括所有特权。可以针对用户、角色和系统扩展初始可继承特权集。还可以将不在可继承特权集中的特权指定给进程。

您可以通过两种方法扩展提供的特权。

- 可以针对用户、角色和系统扩展初始可继承特权集。
- 还可以将不在可继承特权集中的特权显式指定给进程。

按进程指定特权是最精确的添加特权方法。可以通过向某用户指定某种角色，扩展此用户可执行的特权操作的数目。将为该角色指定包含具有已添加特权的命令的权限配置文件。用户承担角色时，便会获取此角色的配置文件 `shell`。当在此角色的 `shell` 中键入权限配置文件中的命令时，将会使用添加的特权来执行这些命令。

还可以将权限配置文件指定给用户而不是用户承担的角色。用户打开配置文件 `shell`（如 `pfksh`）时，便可使用特权执行用户权限配置文件中的命令。在常规 `shell` 中，不使用特权执行命令。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特权集是一种风险性较高的指定特权方法。可继承特权集中的所有特权都位于允许特权集和有效特权集中。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特权。使用直接指定的特权，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。

在某个系统上向初始可继承特权集中添加特权后，登录到此系统的所有用户都会具有更大的基本特权集。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一般用户权限范围的操作。

注 – 不能扩展限制特权集，因为限制特权集最初包括所有特权。

限制用户或角色的特权

通过删除特权，可以防止用户和角色执行特定任务。可以从初始可继承特权集和限制特权集中删除特权。在分配小于缺省特权集的初始可继承特权集或限制特权集之前，应谨慎地测试特权删除操作。通过从初始可继承特权集中删除特权，可以阻止用户登录。如果从限制特权集中删除了特权，传统的 `setuid` 程序可能会失败，因为该程序所需的特权已被删除。

为脚本指定特权

脚本与命令类似，也是可执行文件。因此，在权限配置文件中，可以为脚本添加特权，就像为命令添加特权一样。当分配有权限配置文件的用户或角色在配置文件 `shell` 中执行脚本时，将会使用添加的特权来运行该脚本。如果脚本包含需要特权的命令，则具有已添加特权的命令也必须在指定的权限配置文件中。

可识别特权的程序可以按进程限制特权。对于可识别特权的程序，您的工作是仅为可执行文件指定该程序所需的权限。然后测试此程序，了解程序是否能成功执行其任务。您还需要检查程序是否误用了其特权。

特权和设备

特权模型使用特权保护系统接口，这些接口在超级用户模型中单独受文件权限保护。在具有特权的系统中，文件权限太小，因此无法保护这些接口。`proc_owner` 之类的特权可以覆盖文件权限，然后提供对所有系统的完全访问权限。

因此，在 Oracle Solaris 中，具有设备目录的所有权不足以打开相应设备。例如，不再自动允许 `sys` 组的成员打开 `/dev/ip` 设备。`/dev/ip` 的文件权限是 `0666`，但需要 `net_rawaccess` 特权才能打开该设备。

设备策略受特权控制。`getdevpolicy` 命令可显示每个设备的设备策略。设备配置命令 `devfsadm` 可安装设备策略。`devfsadm` 命令可将特权集与 `open` 绑定以读取或写入设备。有关更多信息，请参见 [getdevpolicy\(1M\)](#) 和 [devfsadm\(1M\)](#) 手册页。

利用设备策略，您可以更加灵活地授予打开设备所需的权限。您可能需要不同于缺省设备策略的特权或更多特权。可以针对设备策略和驱动程序适当地修改特权要求。可以在安装、添加或更新设备驱动程序时修改特权。

`add_drv` 和 `update_drv` 命令用于修改设备策略项以及驱动程序特定的特权。必须运行具有完全特权集的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。

特权和调试

Oracle Solaris 提供了用于调试特权问题的工具。`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 181 页中的“如何确定程序所需的特权”。您也可以使用 `dtrace` 命令。有关更多信息，请参见 [dtrace\(1M\)](#) 手册页。

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 145 页中的“使用 RBAC（任务）”](#)
- [第 175 页中的“使用特权（任务）”](#)

有关 RBAC 的概述，请参见[第 127 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，Oracle Solaris 中的安全属性（参考）](#)。要使用特权，请参见[第 175 页中的“使用特权（任务）”](#)。

使用 RBAC（任务）

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，便可以进一步定制 RBAC 以处理新操作。以下任务列表列出了这些主要任务（包括使用特权）。

任务	说明	参考
使用缺省 RBAC 配置。	在不修改初始安装的情况下查看和使用 RBAC。	第 146 页中的“查看和使用 RBAC 缺省值（任务列表）”
规划、配置和使用 RBAC。	为站点配置 RBAC。	第 153 页中的“初次配置 RBAC（任务列表）”
管理 RBAC。	更新站点的 RBAC 配置。	第 165 页中的“管理 RBAC（任务列表）”
管理和使用特权。	添加和删除用户、角色、系统以及进程的特权。使用特权。查看和调试对特权的使用。	第 175 页中的“使用特权（任务）”

查看和使用 RBAC 缺省值（任务）

在缺省情况下，将为用户指定权限。系统所有用户的权限均在 /etc/security/policy.conf 文件中指定。

查看和使用 RBAC 缺省值（任务列表）

安装 Oracle Solaris 时，您的系统已配置了用户权限和进程权限。在未进行进一步配置的情况下，可使用以下任务列表查看和使用 RBAC。

任务	说明	参考
查看安全属性数据库的内容。	列出系统上的所有授权、权限配置文件以及具有安全属性的命令。	第 146 页中的“如何查看所有已定义的安全属性”
查看权限。	包括列出权限配置文件、授权、特权以及指定的角色。	第 147 页中的“如何查看指定的权限”
承担 root 角色。	初始用户获取管理权限。	第 149 页中的“如何承担角色”
成为管理员。	有多种方法可供已为其指定了管理权限的用户来使用这些权限。	第 150 页中的“如何获取管理权限”

▼ 如何查看所有已定义的安全属性

使用以下命令可列出系统上的所有授权、权限配置文件以及具有安全属性的命令。要列出所有已定义的特权，请参见[第 176 页中的“如何列出系统上的特权”](#)。

1 列出所有授权。

```
% getent auth_attr | more
solaris.:::All Solaris Authorizations::help=AllSolAuthsHeader.html
solaris.account.:::Account Management::help=AccountHeader.html
...
solaris.zone.login.:::Zone Login::help=ZoneLogin.html
solaris.zone.manage.:::Zone Deployment::help=ZoneManage.html
```

2 列出所有权限配置文件。

```
% getent prof_attr | more
All.:::Execute any command as the user or role:help=RtAll.html
Audit Configuration:::Configure Solaris Audit:auths=solaris.smf.value.audit;
help=RtAuditCfg.html
...
Zone Management:::Zones Virtual Application Environment Administration:
help=RtZoneMngmnt.html
Zone Security:::Zones Virtual Application Environment Security:auths=solaris.zone.*,
solaris.auth.delegate;help=RtZoneSecurity.html ...
```

3 列出所有具有安全属性的命令。

```
% getent exec_attr | more
All:solaris:cmd::*:
Audit Configuration:solaris:cmd::/usr/sbin/auditconfig:privs=sys_audit
...
Zone Security:solaris:cmd::/usr/sbin/txzonemgr:uid=0
Zone Security:solaris:cmd::/usr/sbin/zonecfg:uid=0 ...
```

▼ 如何查看指定的权限

使用以下命令可查看 RBAC 指定。要查看可以指定的所有权限，请参见第 146 页中的“如何查看所有已定义的安全属性”。

1 列出授权。

```
% auths
solaris.device.cdrw,solaris.device.mount.removable,solaris.mail.mailq
```

这些授权在缺省情况下会指定给所有用户。

2 列出权限配置文件。

```
% profiles
Basic Solaris User
All
```

这些权限配置文件在缺省情况下会指定给所有用户。

3 列出指定的角色。

```
% roles
root
```

该角色在缺省情况下会指定给初始用户。No roles 表明没有为您指定任何角色。

4 列出缺省 shell 中的特权。

```
% ppriv $$
1234: /bin/csh
flags = <none>
  E: basic
  I: basic
  P: basic
  L: all
```

缺省情况下，将为每个用户指定基本的特权集。限制特权集为所有特权。

```
% ppriv -vl basic
file_link any
    Allows a process to create hardlinks to files owned by a uid
    different from the process' effective uid.
file_read
    Allows a process to read objects in the filesystem.
file_write
    Allows a process to modify objects in the filesystem.
```

```

net_access
    Allows a process to open a TCP, UDP, SDP or SCTP network endpoint.
proc_exec
    Allows a process to call execve().
proc_fork
    Allows a process to call fork1()/forkall()/vfork()
proc_info
    Allows a process to examine the status of processes other
    than those it can send signals to. Processes which cannot
    be examined cannot be seen in /proc and appear not to exist.
proc_session
    Allows a process to send signals or trace processes outside its session.

```

5 列出权限配置文件中各命令的特权。

```

% profiles -l
Basic Solaris User
  /usr/bin/cdda2wav.bin   privs=file_dac_read,sys_devices,
                        proc_priocntl,net_privaddr
  /usr/bin/cdrecord.bin   privs=file_dac_read,sys_devices,
                        proc_lock_memory,proc_priocntl,net_privaddr
  /usr/bin/readcd.bin     privs=file_dac_read,sys_devices,net_privaddr
All
*
```

用户的权限配置文件可以包括使用特定特权运行的命令。基本 Solaris 用户配置文件包括允许用户读取和写入 CD-ROM 的命令。

示例 9-1 列出用户的授权

```

% auths username
solaris.device.cdrw,solaris.device.mount.removable,solaris.mail.mailq

```

示例 9-2 列出用户或角色的权限配置文件

以下命令列出了特定用户的权限配置文件。

```

% profiles jdoe
jdoe:
    Basic Solaris User
    All

```

以下命令列出了 cryptomgt 角色的权限配置文件。

```

% profiles cryptomgt
cryptomgt:
    Crypto Management
    Basic Solaris User
    All

```

以下命令列出了 root 角色的权限配置文件。

```

% profiles root
root:
    All

```



```

Console User
Network Wifi Info
Desktop Removable Media User
Suspend To RAM
Suspend To Disk
Brightness
CPU Power Management
Network Autoconf User
Basic Solaris User

```

示例 9-3 列出用户的指定角色

以下命令列出了特定用户的指定角色。

```
% roles jdoe
root
```

示例 9-4 列出特定命令的用户特权

以下命令列出了一般用户的权限配置文件中的特权命令。

```
% profiles -l jdoe
jdoe:
Basic Solaris User
/usr/bin/cdda2wav.bin   privs=file_dac_read,sys_devices,
proc_prioctl,net_privaddr
/usr/bin/cdrecord.bin   privs=file_dac_read,sys_devices,
proc_lock_memory,proc_prioctl,net_privaddr
/usr/bin/readcd.bin     privs=file_dac_read,sys_devices,net_privaddr
All
*
```

▼ 如何承担角色

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命名服务。

- 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: <Type rolename password>
$
```

su - rolename 命令会将 shell 更改为该角色的配置文件 shell。配置文件 shell 可识别安全属性（例如授权、特权和 set ID 位）。

3 可选验证您现在是否已承担某种角色。

```
$ /usr/bin/whoami
rolename
```

您现在可在此终端窗口中执行角色任务。

4 可选查看角色的功能。

有关过程，请参见第 147 页中的“如何查看指定的权限”。

示例 9-5 承担 root 角色

在以下示例中，初始用户承担 root 角色并列出该角色的 shell 中的特权。

```
% roles
root
% su - root
Password:      <Type root password>
#             Prompt changes to root prompt
# ppriv $$
1200:  pfksh
flags = <none>
      E: all
      I: basic
      P: all
      L: all
```

有关特权的信息，请参见第 138 页中的“特权（概述）”。

▼ 如何获取管理权限

运行配置文件 shell 时，管理权限将生效。在缺省情况下，将为角色帐户指定一个配置文件 shell。角色是特殊帐户，系统为其指定了特定的管理权限，这些权限通常针对相关的一组管理活动（例如查看审计文件）。

在 root 角色中，初始用户具有所有管理权限，即初始用户是超级用户。root 角色可以创建其他角色。

开始之前 要管理系统，您必须具有一般用户没有的权限。如果您不是超级用户，必须为您分配一个角色、管理权限配置文件或特定特权或授权。

- 选择以下方法之一来运行管理命令。

打开终端窗口。

- 成为 root。

```
% su -
Password:      Type the root password
#
```

注 – 无论 root 是用户还是角色，此方法都是可行的。井号 (#) 提示符表明您现在已是超级用户。

- 承担已为您指定的角色。

在以下示例中，您承担了网络管理角色。此角色包括 Network Management（网络管理）权限配置文件。

```
% su - networkadmin
Password:      Type the networkadmin password
$
```

您现已位于配置文件 shell 中。在该 shell 中，可以运行 snoop、route、dladm 以及其他命令。有关配置文件 shell 的更多信息，请参见第 136 页中的“配置文件 Shell 和 RBAC”。

提示 – 使用第 147 页中的“如何查看指定的权限”中介绍的步骤查看角色的功能。

- 使用 pfbash 命令创建通过管理权限运行的 shell。

例如，以下命令集允许您在 pfbash shell 中检查网络包：

```
% pfbash
$ anoop
```

如果没有为您分配 net_observability 特权，则 snoop 命令将失败，并出现一条类似于以下内容的错误消息：snoop: cannot open "net0": Permission denied。如果直接或者通过权限配置文件或角色为您指定特权，该命令将运行成功。此外，您还可以在此 shell 中运行其他特权命令。

- 使用 pfexec 命令创建通过管理权限运行的进程。

结合使用权限配置文件的特权命令名称来运行 pfexec 命令。例如，以下命令允许您检查网络包：

```
% pfexec snoop
```

应用于 pfexec 的特权限制与应用于 pfbash 的特权限制相同。但要运行其他特权命令，必须在键入该特权命令之前再次键入 pfexec。

示例 9-6 高速缓存验证以便于使用角色

在以下示例中，管理员会配置一个角色来管理网络，但为方便使用该角色，会高速缓存用户的验证。首先，管理员创建并指定角色。

```
# roleadd -K roleauth=user -P "Network Management" netmgt
# usermod -R +netmgt jdoe
```

当切换到该角色时，如果 jdoe 使用 -c 选项，则在显示 snoop 输出之前需要输入口令。

```
% su - netmgt -c snoop options
Password:
```

```
    snoop output
```

如果没有高速缓存验证，jdoe 将立即重新运行该命令，并出现输入口令提示。

管理员配置 pam.conf 文件来高速缓存验证，以便只需要在初次运行时输入口令，以后不再需要输入口令，直到超过某段时间之后。管理员会将所有 pam.conf 定制堆栈放在文件的结尾处。

```
# vi /etc/pam.conf
...
#
## Cache authentication for switched user
#
su      auth required          pam_unix_cred.so.1
su      auth sufficient        pam_tty_tickets.so.1
su      auth requisite         pam_authtok_get.so.1
su      auth required          pam_dhkeys.so.1
su      auth required          pam_unix_auth.so.1
```

创建项之后，管理员将对这些项进行检查，确定是否出现拼写、缺失或重复问题。

需要整个 su 堆栈。pam_tty_tickets.so.1 模块提供高速缓存。有关 PAM 的更多信息，请参见 [pam.conf\(4\)](#) 手册页以及 [第 15 章，使用 PAM](#)。

将 su PAM 堆栈添加到 pam.conf 文件之后，对于 netmgt 角色，当运行一系列命令时仅提示一次输入口令。

```
% su - netmgt -c snoop options
Password:
```

```
    snoop output
```

```
% su - netmgt -c snoop options
    snoop output
...
```

为站点定制 RBAC（任务）

RBAC 的初始配置包括创建可以承担特定角色的用户，创建角色并将这些角色指定给相应的用户。

初次配置 RBAC（任务列表）

使用以下任务列表可在站点上规划并初步实现 RBAC。一些任务是按顺序排序的。

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 153 页中的“如何规划 RBAC 实现”
2. 配置可以承担角色的用户。	确保存在可以承担管理角色的用户。	《Oracle Solaris 管理：常见任务》中的“设置和管理用户帐户（任务列表）”
3. 创建角色。	创建角色并将角色指定给用户	第 155 页中的“如何创建角色” 第 157 页中的“如何指定角色”
（建议）审计角色操作。	预选包括记录角色操作的审计事件的审计类。	第 159 页中的“如何审计角色”
创建或更改权限配置文件。	创建权限配置文件。或修改权限配置文件中的安全属性或补充权限配置文件。 将特权添加到命令中。	第 159 页中的“如何创建或更改权限配置文件” 示例 9-14
确保传统应用程序安全	为传统应用程序启用 set ID 权限。脚本可以包含具有 set ID 的命令。传统应用程序可检查授权（如果适用）。	第 161 页中的“如何为传统应用程序添加 RBAC 属性” 示例 9-16
排除安全属性指定故障。	针对有关指定的安全属性为何不能用于用户、角色或进程的原因进行调试。	第 162 页中的“如何排除 RBAC 和特权指定故障”

▼ 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。

注 – 在 `/etc/security/policy.conf` 文件中指定缺省权限。

1 了解 RBAC 基本概念。

请阅读第 127 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 管理做法大不相同。要在开始实现之前熟悉 RBAC 概念, 请参见第 10 章, [Oracle Solaris 中的安全属性 \(参考\)](#)。

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提供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。尽管可以按原样使用安装的 RBAC 配置, 但可能需要对其进行定制, 使其支持您的安全策略。

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:

- **root 作为角色**—缺省情况下提供该方法。它可以防止任何用户以 root 身份登录。相反, 用户必须在承担 root 角色之前通过为他们指定的登录信息进行登录。
- **独立角色**—此方法创建基于提供的权限配置文件的角色。可以根据职责级别、任务范围和任务类型指定角色。例如, System Administrator (系统管理员) 角色可以执行超级用户可执行的许多任务, 而 Network IPsec Management (网络 IPsec 管理) 角色可以管理 IPsec。

您还可以将安全职责与其他职责分隔开。User Management (用户管理) 角色可以创建用户, 而 User Security (用户安全) 角色可以指定安全属性 (例如角色和权限配置文件)。但 User Security (用户安全) 角色无法创建用户, User Management (用户管理) 角色无法将权限配置文件指定给用户。

- **没有 root 角色**—此方法需要您更改系统的缺省配置。在此配置中, 知道 root 口令的任何用户都可以登录和修改系统。您无法区分充当超级用户的用户。

4 确定适用于组织的角色。

请查看建议的角色的功能及其缺省权限配置文件。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。

要进一步检查权限配置文件, 请执行以下操作之一:

- 要获取系统上可用的权限配置文件, 请使用 `getent prof_attr` 命令。
- 在本指南中, 请参阅第 185 页中的“[权限配置文件](#)”以了解某些典型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。您可以定制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。

注 – Media Backup (介质备份) 或 Media Restore (介质恢复) 权限配置文件可提供访问整个根文件系统的权限。因此，这些权限配置文件仅指定给相应的可信用户。或者，您可以选择不指定这些权限配置文件。缺省情况下，仅信任 `root` 角色可以进行备份和恢复。

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，请检查该角色的原始权限配置文件是否适用于已指定给该角色的用户。对新的权限配置文件进行排序，以便命令能够使用所需的特权执行。有关排序的信息，请参见第 187 页中的“搜索指定安全属性的顺序”。

6 确定应将哪些用户指定给哪些角色。

根据 `least privilege` (最小特权) 原则，应将用户指定给适合该用户信任级别的角色。如果禁止用户执行用户无需执行的任务，可以减少潜在的问题。

▼ 如何创建角色

可以在本地以及 LDAP 系统信息库中创建角色。

开始之前 要创建角色并指定其初始口令，必须为您指定 `User Management` (用户管理) 权限配置文件。要将安全属性指定给角色，必须为您指定 `User Security` (用户安全) 权限配置文件。

1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 要创建角色，请使用 `roleadd` 命令。

该命令的 RBAC 参数如下所示：

```
# roleadd [-e expire] [-f inactive] [-s shell] [-m] [-S repository] \
[-A authorization-list] -K key=value rolename
```

- `-e expire` 角色到期的日期。使用该选项可以创建临时角色。
- `-f inactive` 连续两次使用同一角色之间的时间间隔所允许的最大天数。当超出 `inactive` 值时，无法使用该角色。缺省值为 `0`，没有失效日期。
- `-m` 在缺省位置为 `rolename` 创建起始目录。

<code>-s shell</code>	<code>rolename</code> 的登录 shell。此 shell 必须是配置文件 shell。有关配置文件 shell 的列表，请参见 pfexec(1) 手册页。
提示 – 如 <code>ls /usr/bin/pf*sh</code> 中一样，您还可以列出系统 <code>/usr/bin</code> 目录中的配置文件 shell。	
<code>-S repository</code>	<code>files</code> 或 <code>ldap</code> 之一。缺省值为本地文件。
<code>-A authorization-list</code>	是一个或多个授权（以逗号分隔）。有关授权的列表，请参见 <code>/etc/security/auth_attr</code> 文件。
<code>-K key=value</code>	<code>key=value</code> 对。可以重复该选项。提供以下密钥： <code>audit_flags</code> 、 <code>auths</code> 、 <code>profiles</code> 、 <code>project</code> 、 <code>defaultpriv</code> 、 <code>limitpriv</code> 、 <code>lock_after_retries</code> 和 <code>roleauth</code> 。有关设置值所需的密钥、密钥值以及授权的信息，请参见 user_attr(4) 手册页。
<code>rolename</code>	新角色的名称。有关可接受的字符串的限制，请参见 roleadd(1M) 手册页。

提示 – 当角色的名称反映权限配置文件的名称时，就可以轻松地了解该角色的目的。例如，将 **Audit Review**（审计查看）权限配置文件指定给 `auditreview` 角色时，以使得该角色可以读取、过滤和归档审计记录。

例如，以下命令可以创建一个本地 User Administrator（用户管理员）角色和一个起始目录。

```
# roleadd -c "User Administrator role, local" -s /usr/bin/pfbash \
-m -K profiles="User Security,User Management" useradm
80 blocks
# ls /export/home/useradm
local.cshrc      local.login      local.profile
```

3 为该角色创建初始口令。

```
# passwd -r files useradmPassword:      <Type useradm password>
Confirm Password:    <Retype useradm password>
#
```

注 – 通常，可以将一种角色帐户指定给多个用户。因此，管理员通常都会创建一个角色口令，并单独为用户提供角色口令。

4 要将角色指定给某个用户，请运行 `usermod` 命令。

有关过程，请参见第 157 页中的“如何指定角色”和示例 9-10。

示例 9-7 在 LDAP 系统信息库中创建 User Administrator（用户管理员）角色

在以下示例中，管理员站点使用 LDAP 系统信息库。通过运行以下命令，管理员可以在 LDAP 中创建一个 User Administrator（用户管理员）角色。

```
# roleadd -c "User Administrator role, LDAP" -s /usr/bin/pfbash \
-m -S ldap -K profiles="User Security,User Management" useradm
```

示例 9-8 创建角色以实现职责分离

在以下示例中，管理员站点使用 LDAP 系统信息库。通过运行以下命令，管理员可以创建两个角色。usermgt 角色可以创建用户，为其提供起始目录，指定初始口令以及执行其他非安全任务。usersec 角色无法创建用户，但可以更改用户口令以及其他 RBAC 属性。

```
# roleadd -c "User Management role, LDAP" -s /usr/bin/pfbash \
-m -S ldap -K profiles="User Management" usermgt
# roleadd -c "User Security role, LDAP" -s /usr/bin/pfbash \
-m -S ldap -K profiles="User Security" usersec
```

示例 9-9 创建 Device and File Security（设备和文件安全）角色

在以下示例中，管理员将为此系统创建 Device and File Security（设备和文件安全）角色：

```
# roleadd -c "Device and File System Security admin, local" -s /usr/bin/pfbash \
-m -K profiles="Device Security,File System Security" devflsec
```

▼ 如何指定角色

此过程会将角色指定给用户、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。

开始之前 您已添加了一个角色并为该角色指定了一个口令，如第 155 页中的“如何创建角色”中所述。

要修改用户的大部分安全属性，必须为您指定 User Security（用户安全）权限配置文件。只有超级用户才能修改用户的审计标志。要修改其他属性，必须为您指定 User Management（用户管理）权限配置文件。

1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 将角色指定给用户。

```
usermod [-S repository] [RBAC-arguments] login
```

例如，将角色给本地用户。

```
# usermod -R +useradm jdoe-local
```

有关 `usermod` 命令的选项的信息，请参见 `usermod(1M)` 手册页或第 155 页中的“如何创建角色”中步骤 2 的介绍。

3 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
```

示例 9-10 创建和指定角色以管理加密

在以下示例中，LDAP 网络的管理员会创建一个角色以管理加密框架，并将该角色指定给 UID 1111。管理员重新启动 `nscd` 守护进程以使指定生效。

```
# roleadd -c "Cryptographic Services manager" \  
-g 14 -m -u 104 -s /usr/bin/pfksh \  
-S ldap -K profiles="Crypto Management" cryptmgt  
# passwd cryptmgt  
New Password:      <Type cryptmgt password>  
Confirm password:  <Retype cryptmgt password>  
# usermod -u 1111 -R +cryptmgt  
# svcadm restart system/name-service-cache
```

具有 UID 1111 的用户将登录，然后承担该角色并显示指定的安全属性。

```
% su - cryptmgt  
Password:      <Type cryptmgt password>  
Confirm Password:  <Retype cryptmgt password>  
$ profiles -l  
    Crypto Management  
        /usr/bin/kmfcfg          euid=0  
        /usr/sbin/cryptoadm      euid=0  
        /usr/sfw/bin/CA.pl       euid=0  
        /usr/sfw/bin/openssl     euid=0  
  
$
```

有关加密框架的信息，请参见第 11 章，加密框架（概述）。要管理该框架，请参见第 218 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。116:AUE_PFEXEC:execve(2) with pfexec enabled:ps,ex,ua,as 审计事件捕获角色操作。通过预选 as、ex、ps 或 ua 类之一，来审计角色操作。

开始之前 要配置审计，必须为您指定 Audit Configuration（审计配置）权限配置文件。要启用或刷新审计服务，必须为您指定 Audit Control（审计控制）权限配置文件。

1 包括审计规划中的角色审计。

有关规划信息，请参见第 27 章，规划审计。

2 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

3 预选 as、ex、ps 或 ua 类之一。

- 如果已启用审计服务，请查看预选的类。

```
# auditconfig -getflags
```

如果预选 as、ex、ps 或 ua 类之一，将审计角色操作。否则，请将其中一个类添加到现有类中。

```
# auditconfig -setflags existing preselections,as
```

- 如果尚未启用审计，请预选一个可用于审计角色操作的类。

```
# auditconfig -setflags as
```

在该示例中，管理员选择了 as 类。该类包括其他审计事件。要查看一个类中包括的审计事件，请使用 auditrecord 命令，如示例 28-25 中所示。

4 启用或刷新审计服务。

```
# audit -s
```

▼ 如何创建或更改权限配置文件

当提供的权限配置文件不包含所需的安全属性集合时，您可以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 135 页中的“RBAC 权限配置文件”。

创建新的权限配置文件的最简单方法是复制并修改现有的权限配置文件。

开始之前 要创建或更改权限配置文件，必须为您指定 File Security（文件安全）权限配置文件。

1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 通过现有的配置文件创建新的权限配置文件。

```
# profiles [-S repository] existing-profile-name
```

此时会提示您输入新名称。现有权限配置文件的内容会被复制到新的配置文件中。

3 继续修改新的权限配置文件。

添加或删除补充权限配置文件、授权以及其他安全属性，如以下示例中所示。

示例 9-11 通过现有的配置文件创建新的权限配置文件

在以下示例中，管理员在 LDAP 系统信息库中定制 Console User（控制台用户）权限配置文件。

```
# profiles -S ldap Console User
New name: ExampleCo Console User
ExampleCo Console User >
Description > Manage MyCompany Systems as the Console User
Help > ExCoConsUser.html
```

管理员设置此权限配置文件的 roleauth 属性。

```
roleauth=yes
```

示例 9-12 从权限配置文件中删除基本特权

在以下示例中，详尽测试后，安全管理员从分配有 SunRayUser 权限配置文件的所有用户中删除基本特权。防止这些用户使用 proc_session 特权。即，这些用户无法检查用户当前会话之外的任何进程。

```
$ profiles -K defaultpriv=basic,!proc_session SunRayUser
```

示例 9-13 从权限配置文件的限制特权集中删除特权

在以下示例中，详尽测试后，安全管理员从分配有 SunRayUser 权限配置文件的所有用户中删除限制特权。此删除行为可防止这些用户查看其他用户的进程。

```
$ profiles -K limitpriv=all,!proc_session SunRayUser
```

示例 9-14 将特权添加到命令中

在以下示例中，安全管理员在权限配置文件添加某应用程序的特权。此应用程序能够识别特权。

```
# profiles -p SiteApp
profiles:SiteApp> set desc="Site application"
profiles:SiteApp> add cmd=/opt/site-app/bin/site-cmd
profiles:SiteApp:site-cmd> add privs=proc_fork,proc_taskid
profiles:SiteApp:site-cmd> end
profiles:SiteApp> exit
```

要进行验证，管理员可以选择 `site-cmd`。

```
# profiles -p SiteApp "select cmd=/opt/site-app/bin/site-cmd; info;end"
Found profile in files repository.
  id=/opt/site-app/bin/site-cmd
  privs=proc_fork,proc_taskid
```

另请参见 要排除安全属性分配故障，请参见第 162 页中的“如何排除 RBAC 和特权指定故障”。有关背景的信息，请参见第 187 页中的“搜索指定安全属性的顺序”。

▼ 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属性的传统应用程序。

开始之前 要创建权限配置文件，必须为您指定 Information Security（信息安全）或 Rights Management（权限管理）权限配置文件。要指定权限配置文件，必须为您指定 User Security（用户安全）权限配置文件。

1 向实现传统应用程序的命令添加安全属性。

向传统应用程序添加安全属性的方式与为任何命令添加安全属性的方式相同。您必须将具有安全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` 安全属性。有关该过程的详细信息，请参见第 159 页中的“如何创建或更改权限配置文件”。

a. 为传统应用程序创建新的权限配置文件。

有关步骤，请参见第 159 页中的“如何创建或更改权限配置文件”。

b. 添加包含所需安全属性的命令。

有关示例，请参见示例 9-14。

- 2 在角色的配置文件列表中加入权限配置文件。
要为角色指定权限配置文件，请参见[示例 9-10](#)。

示例 9-15 向脚本中的命令添加安全属性

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，此命令便会以安全属性运行。

示例 9-16 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，必须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.system.date` 授权，需要检查以下字符串：

- `solaris.system.date`
- `solaris.system.*`
- `solaris.*`

如果您要编写程序，请使用函数 `getauthattr()` 对授权进行测试。

▼ 如何排除 RBAC 和特权指定故障

对于用户或角色的进程为何没有使用指定的安全属性运行，存在以下几种影响因素。

- 安全属性拼写错误。拼写错误的授权失败而不会有提示。
- 用户或角色未使用包括指定的命名服务。
- 您所需的指定并不是该属性的第一个指定。

搜索并随后在验证时指定用户或角色安全属性的顺序决定了哪些指定能够成功。授权是例外。在搜索期间，会累积指定给用户或角色的授权。与之相反，权限配置文件中的特权指定和安全属性的指定与搜索紧密相关。将使用第一个指定，而后续指定都将被忽略。

- 命令不在配置文件 shell 中运行。

开始之前 您必须是 root 角色。

1 验证并重新启动命名服务。

- a. 验证用户或角色的安全指定是否位于在系统上启用的命名服务中。

- b. 重新启动名称服务高速缓存 `svc:/system/name-service/cache`。

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命名服务。

2 确定指定安全属性的位置。

将安全属性用作 `userattr -v` 命令的值。例如，以下命令指明了对用户 `jdoe` 指定的安全属性以及执行指定操作的位置：

```
# userattr -v audit_flags jdoe      Modifications to the system defaults
user_attr: fw:no
# userattr -v auths jdoe            Assigned authorizations
solaris.admin.wusb.read,solaris.device.cdrw,solaris.device.mount.removable,
solaris.mail.mailq,solaris.profmgr.read,solaris.smf.manage.audit,
solaris.smf.value.audit
# userattr -v audit_flags jdoe      Modifications to audit preselection mask
# userattr -v auths jdoe            Assigned authorizations
# userattr -v defaultpriv jdoe      Modifications to basic user privileges
# userattr -v limitpriv jdoe        Modifications to limit privileges
# userattr -v lock_after_retries jdoe Automatic lockout attribute
# userattr -v profiles jdoe        Assigned rights profiles
user_attr: Audit Review,Stop
# userattr roles jdoe              Assigned roles
user_attr : cryptomgt,infosec
```

3 对于所创建的权限配置文件，请验证是否将相应的安全属性指定给命令。

例如，为能够成功运行，某些命令要求 `uid=0`，而非 `euid=0`。一些命令的各个方面都需要授权。

4 如果安全属性对用户不可用，请检查以下各项。

- a. 检查是否将安全属性直接指定给用户。

请使用 `userattr` 命令。

b. 如果未直接指定安全属性，请检查直接指定给用户的权限配置文件。

i. 按顺序检查权限配置文件列表中的安全属性指定。

该列表中最早权限配置文件中的属性值即为用户可以使用的值。如果此值不正确，请在该权限配置文件中更改该值，或者对配置文件的列表重新进行排序。

对于特权命令，请检查是否在 `defaultpriv` 关键字中指定了特权。该指定是对特定命令的特权的补充。

ii. 如果未列出任何属性指定，请检查为用户指定的角色。

如果属性已指定给一个角色，则该用户只有承担该角色才能获取安全属性。如果属性已指定给多个角色，则列表中最早角色的指定是有效指定。如果此值不正确，请将正确的值指定给列表中的第一个角色，或者对角色指定重新进行排序。

5 如果已将特权直接指定给用户角色，请检查失败的命令是否需要授权才能成功运行。

注—一些命令的各个方面都需要授权。最佳做法是指定包括管理命令的权限配置文件，而非直接指定特权。

查看包括管理命令的权限配置文件。如果存在包括授权的权限配置文件，请将该权限配置文件指定给用户，而不是简单的指定特权。将该权限配置文件放在所有包括该命令的其他权限配置文件之前。

6 如果一个命令针对某用户持续运行失败，请检查以下项。

a. 验证用户是否正在配置文件 `shell` 中执行该命令。

必须在配置文件 `shell` 中执行管理命令。要更正用户错误，可以将配置文件 `shell` 指定为用户的登录 `shell`。或者，您可以提醒用户在配置文件 `shell` 中运行管理命令。

b. 检查直接指定给用户的任何安全属性是否阻碍了该命令的成功运行。

特别是，检查用户的 `defaultpriv` 和 `limitpriv` 属性的值。

c. 确定包括该命令的权限配置文件或角色。

i. 按顺序检查权限配置文件列表中包含安全属性的命令。

权限配置文件列表中最早的值即为用户可以使用的值。如果此值不正确，请在该权限配置文件中更改该值，或者对配置文件的列表重新进行排序。

特别是，检查配置文件的 `defaultpriv` 和 `limitpriv` 属性的值。

ii. 如果未列出任何属性指定，请检查为用户指定的角色。

如果该命令已指定给某角色，则用户只有承担该角色才能获取安全属性。如果属性已指定给多个角色，则列表中最早角色的指定是有效指定。如果此值不正确，请将正确的值指定给列表中的第一个角色，或者对角色指定重新进行排序。

7 如果一个命令针对某角色运行失败，请检查以下项。

管理命令需要特权才能成功运行。一些命令的各个方面都需要授权。最佳做法是指定包括管理命令的权限配置文件。

a. 检查直接指定给角色的任何安全属性是否阻碍了该命令的成功运行。

特别是，检查角色的 defaultpriv 和 limitpriv 属性的值。

b. 按顺序检查权限配置文件列表中包含安全属性的命令。

权限配置文件列表中最早的值即为用户可以使用的值。如果此值不正确，请在该权限配置文件中更改该值，或者对配置文件的列表重新进行排序。

管理 RBAC（任务）

在已配置并使用 RBAC 后，请使用以下过程维护并修改系统上的 RBAC。

管理 RBAC（任务列表）

以下任务列表列出了有关在初步实现基于角色的访问控制 (role-based access control, RBAC) 后维护 RBAC 的过程。

任务	说明	参考
更改角色口令。	授权的用户或角色更改另一个角色的口令。	第 166 页中的“如何更改角色的口令”
修改角色的指定权限。	修改角色的安全属性。	第 167 页中的“如何更改角色的安全属性” 示例 9-19
修改用户的权限。	为一般用户添加安全属性，或删除这些安全属性。	第 168 页中的“如何更改用户的 RBAC 属性” 示例 9-24 示例 9-12
修改权限配置文件中的用户权限。	指定权限配置文件中的安全属性，例如审计标志和缺省特权。	示例 9-21 示例 9-13

任务	说明	参考
创建受限制的配置文件 shell。	防止用户或角色完全访问软件中的所有命令。	第 171 页中的“如何将管理员限于显式指定的权限”
从系统中删除缺省权限。	创建一个系统供在特殊时候使用。	示例 9-25
限制用户的特权。	限制用户的基本或限制特权集。	示例 9-21
允许用户提供相应的用户口令，以承担角色。	修改用户的安全属性，以便该用户的口令能够使该用户通过角色的验证。此行为与 Linux 角色行为类似。	第 172 页中的“如何让用户使用自己的口令承担角色”
将 root 更改为用户。	在淘汰系统之前，将 root 角色更改为用户。	第 173 页中的“如何将 root 角色更改为用户”

以上这些过程管理用户、角色以及权限配置文件的安全属性。有关基本的用户管理过程，请参阅《Oracle Solaris 管理：常见任务》中的第 2 章“管理用户帐户和组（概述）”。

▼ 如何更改角色的口令

开始之前 您必须是 root 角色。

- 运行 **passwd** 命令。

```
# passwd [-r naming-service] target-rolename
```

-r naming-service 将口令更改应用于 files 或 ldap 系统信息库。缺省的系统信息库为 files。如果没有指定系统信息库，所有系统信息库中的口令都将发生更改。

target-rolename 要修改的现有角色的名称。

有关更多命令选项，请参见 [passwd\(1\)](#) 手册页。

示例 9-17 更改角色的口令

在以下示例中，root 角色将更改本地 devmgt 角色的口令。

```
# passwd -r files devmgt
New password:      Type new password
Confirm password:  Retype new password
```

在以下示例中，root 角色将更改 LDAP 目录服务中 devmgt 角色的口令。

```
# passwd -r ldap devmgt
New password:      Type new password
Confirm password:  Retype new password
```

在以下示例中，root 角色将更改文件和 LDAP 中 devmgt 角色的口令。

```
# passwd devmgt
New password:      Type new password
Confirm password:  Retype new password
```

▼ 如何更改角色的安全属性

开始之前 您必须分配有 User Security（用户安全）权限配置文件之后，您才能更改角色的安全属性（除了角色的口令和审计标志）。角色属性包括权限配置文件和授权。要指定审计标志或更改角色的口令，必须采用 root 角色。

注-要更改口令，请参见第 166 页中的“如何更改角色的口令”。

1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 使用 rolemod 命令。

此命令可修改在本地命名服务或 LDAP 中定义的角色属性。-A、-P 和 -R 选项的值可以通过 - 或 ++ 进行修改。- 表示从当前指定的值中减去该值。++ 表示将该值与当前指定的值相加。

有关 rolemod 命令的更多信息，请参见以下内容：

- 如果要简单了解，请参见第 155 页中的“如何创建角色”中有关 roleadd 命令的介绍。
- 有关此命令的所有参数，请参见 rolemod(1M) 手册页。
- 有关 -K 选项的关键值列表，请参见 user_attr(4) 手册页。

以下命令可以替换 LDAP 系统信息库中 devmgt 角色的指定权限配置文件。

```
$ rolemod -P "Device Management,File Management" -S ldap devadmin
```

示例 9-18 更改本地角色的安全属性

在以下示例中，管理员将修改 prtmgmt 角色以包括 VSCAN Management（VSCAN 管理）权限配置文件。

```
$ rolemod -c "Handles printers and virus scanning" \
-P "Printer Management,VSCAN Management,All" prtmgmt
```

这些权限配置文件将添加到通过 policy.conf 文件授予的配置文件。

示例 9-19 将特权直接指定给角色

在以下示例中，安全管理员会向 `sysstime` 角色授予一个非常特定的特权，该特权可以影响系统时间。

```
$ rolemod -K priv=proc_clock_highres sysstime
```

`priv` 关键字的值始终都位于角色进程的特权列表中。

▼ 如何更改用户的 RBAC 属性

用户属性包括登录 shell、权限配置文件和角色。为用户提供管理权能的最安全方法是将角色指定给用户。有关讨论，请参见第 137 页中的“直接指定安全属性时的安全注意事项”。

开始之前 您必须分配有 User Security（用户安全）权限配置文件之后，您才能更改用户的安全属性（除了用户的口令和审计标志）。要指定审计标志或更改角色的口令，必须采用 `root` 角色。要更改其他用户属性，必须为您指定 User Management（用户管理）权限配置文件。

1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 使用 `usermod` 命令。

该命令可以修改在本地命名服务或 LDAP 命名服务中定义的用户属性。该命令的 RBAC 参数类似于 `useradd` 命令的参数，如 [user_attr\(4\)](#) 手册页和 [示例 9-23](#) 中所述。

在以下示例中，为 LDAP 用户指定了 `devmgt` 角色。该角色将替换所有之前的角色指定。`devmgt` 角色必须存在于 LDAP 命名服务中。

```
$ usermod -R devmgt -S ldap jdoe-ldap
```

在以下示例中，该角色将添加到所有之前的角色指定中。

```
$ usermod -R +devmgt -S ldap jdoe-ldap
```

示例 9-20 将角色指定给本地用户。

在本示例中，用户 `jdoe` 现在可以承担 System Administrator（系统管理员）`sysadmin` 角色。

```
$ userattr roles jdoe
secdevice
$ usermod -R secdevice,sysadmin jdoe
```

```
$ userattr roles jdoe
secdevice,sysadmin
```

示例 9-21 从用户的限制特权集中删除特权

在以下示例中，防止所有源自 `jdoe` 初始登录的会话使用 `sys_linkdir` 特权。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
$ userattr limitpriv jdoe
all,!sys_linkdir
```

示例 9-22 创建可以管理 DHCP 的用户

在本示例中，安全管理员将在 LDAP 中创建一个用户。登录时，`jdoe-dhcp` 用户能够管理 DHCP。

```
# useradd -P "DHCP Management" -s /usr/bin/pfbash -S ldap jdoe-dhcp
```

由于已为用户指定了 `pfbash` 作为登录 shell，因此 DHCP Management（DHCP 管理）权限配置文件中的安全属性对用户缺省 shell 中的用户可用。

示例 9-23 将授权直接指定给用户

在本示例中，安全管理员将创建一个可以控制屏幕亮度的本地用户。

```
# useradd -c "Screened JDoe, local" -s /usr/bin/pfbash \
-A solaris.system.power.brightness jdoe-scr
```

该授权将添加到用户的现有授权指定中。

示例 9-24 将特权直接指定给用户

在本示例中，安全管理员会向 `jdoe` 用户授予一个非常特定的特权，该特权可以影响系统时间。

```
$ usermod -K defaultpriv=basic,proc_clock_highres jdoe
```

`defaultpriv` 关键字的值将替换现有的值。因此，对于要保留 `basic` 特权的用户，指定了值 `basic`。在缺省配置中，所有用户都拥有基本特权。

▼ 如何将用户限于桌面应用程序

仅可以将 Oracle Solaris 用户限于桌面访问。

开始之前 您必须是 root 角色。

- 1 为用户指定一个配置文件 shell，作为登录 shell。

例如，可以将 pfbash shell 指定给用户。

```
# usermod -s /usr/bin/pfbash username
```

所有用户进程现在都受 RBAC 控制。

- 2 创建允许用户在 Oracle 桌面上运行基本 applet 的权限配置文件。

以下命令可以创建权限配置文件。end 命令表明添加的命令不需要安全属性。要在 LDAP 系统信息库中创建权限配置文件，请使用 -S ldap 选项。

```
# profiles -p "Desktop Applets"
profiles:Desktop Applets> set desc="Can use basic desktop applications"
profiles:Desktop Applets> add cmd=/usr/bin/nautilus;end
profiles:Desktop Applets> add cmd=/usr/bin/dbus-launch;end
profiles:Desktop Applets> add cmd=/usr/lib/dbus-daemon;end
profiles:Desktop Applets> add cmd=/usr/lib/clock-applet;end
profiles:Desktop Applets> add cmd=/usr/lib/gconfd-2;end
profiles:Desktop Applets> add cmd=/usr/lib/gvfsd;end
profiles:Desktop Applets> add cmd=/usr/lib/gvfsd-metadata;end
profiles:Desktop Applets> add cmd=/usr/lib/gvfsd-trash;end
profiles:Desktop Applets> add cmd=/usr/lib/gvfs-hal-volume-monitor;end
profiles:Desktop Applets> add cmd=/usr/lib/gnome-pty-helper;end
profiles:Desktop Applets> add cmd=/usr/lib/utmp_update;end
profiles:Desktop Applets> add cmd=/usr/bin/sh;end
profiles:Desktop Applets> add cmd=/usr/bin/bash;end
profiles:Desktop Applets> add cmd=/usr/bin/csh;end
profiles:Desktop Applets> add cmd=/usr/bin/ksh;end
profiles:Desktop Applets> commit
profiles:Desktop Applets> exit
```

- 3 验证权限配置文件中是否包含正确的项。

查看各项是否出现错误，例如拼写、缺失或重复错误。

```
# profiles -p "Desktop Applets" info
Found profile in files repository.
name=Desktop Applets
desc=Can use basic desktop applications
cmd=/usr/bin/nautilus
cmd=/usr/bin/dbus-launch
cmd=/usr/lib/dbus-daemon
cmd=/usr/lib/clock-applet
cmd=/usr/lib/gconfd-2
cmd=/usr/lib/gvfsd
cmd=/usr/lib/gvfsd-metadata
cmd=/usr/lib/gvfsd-trash
cmd=/usr/lib/gvfs-hal-volume-monitor
```

```
cmd=/usr/lib/gnome-pty-helper
cmd=/usr/lib/utmp_update
cmd=/usr/bin/sh
cmd=/usr/bin/bash
cmd=/usr/bin/csh
cmd=/usr/bin/ksh
```

提示 – 您可以为一个应用程序或一类具有桌面图标的应用程序创建一个权限配置文件。然后，将 Desktop Applets（桌面 Applet）作为补充权限配置文件添加到该新的权限配置文件中。结合使用这些权限配置文件可允许用户使用适当的桌面应用程序。

4 将 Desktop Applets（桌面 Applet）和 Stop（停止）权限配置文件指定给用户。

```
# usermod -P "Desktop Applets,Stop" username
```

该用户没有 Basic Solaris User（基本 Solaris 用户）或 Console User（控制台用户）权限配置文件。因此，该用户不能运行 Desktop Applets（桌面 Applet）权限配置文件中的命令以外的任何命令。例如，用户无权访问终端窗口。

有关更多信息，请参见第 185 页中的“权限配置文件”、第 187 页中的“搜索指定安全属性的顺序”和《Trusted Extensions 配置和管理》中的“如何限定用户仅使用桌面应用程序”。

usermod 命令可以修改在本地命名服务或 LDAP 中定义的用户属性。有关此命令的参数，请参见 [usermod\(1M\)](#) 手册页。

▼ 如何将管理员限于显式指定的权限

您可以通过下列两种方式限制角色或用户，使其执行有限数量的管理操作。

- 可以使用 Stop（停止）权限配置文件。
Stop（停止）权限配置文件是创建受限 shell 的最简单方式。不会参考在 policy.conf 文件中指定的授权和权限配置文件。在缺省配置中，并没有为角色或用户指定 Basic Solaris User（基本 Solaris 用户）权限配置文件、Console User（控制台用户）权限配置文件或 solaris.device.cdrw 授权。
- 可以修改某系统的 policy.conf 文件，并要求角色或用户使用该系统执行管理任务。

开始之前 您必须是 root 角色。

- 添加 Stop（停止）权限配置文件，作为指定的配置文件列表中的最后一个配置文件。例如，可以限制 auditrev 角色，使其仅执行审计查看。

```
# rolemod -P "Audit Review,Stop" auditrev
```

由于 `auditrev` 角色不具有 `Console User`（控制台用户）权限配置文件，因此审计者无法关闭系统。由于该角色不包含 `solaris.device.cdrw` 授权，因此审计者无法读取或写入 CD-ROM 驱动器。由于该角色不具有 `Basic Solaris User`（基本 Solaris 用户）权限配置文件，因此该角色不能运行 `Audit Review`（审计查看）权限配置文件中的命令以外的任何命令。例如，不能运行 `ls` 命令。该角色使用 `File Browser`（文件浏览者）查看审计文件。

有关更多信息，请参见第 185 页中的“权限配置文件”和第 187 页中的“搜索指定安全属性的顺序”。

`rolemod` 命令可修改在本地命名服务或 LDAP 中定义的角色属性。有关此命令的参数，请参见 [rolemod\(1M\)](#) 手册页。RBAC 参数列表与 `roleadd` 命令列表相似，如第 155 页中的“如何创建角色”中所述。

示例 9-25 修改系统以使权限仅对其用户可用

在以下示例中，管理员将创建一个仅可以用来管理网络的系统。管理员将从 `policy.conf` 文件中删除 `Basic Solaris User`（基本 Solaris 用户）权限配置文件和 `solaris.device.cdrw` 授权。不删除 `Console User`（控制台用户）权限配置文件。在生成的 `policy.conf` 文件中，受影响的行如下所示：

```
...
#AUTHS_GRANTED=solaris.device.cdrw
#PROFS_GRANTED=Basic Solaris User
CONSOLE_USER=Console User
...
```

只有显式为其指定了授权、命令或权限配置文件的用户才能使用该系统。登录后，已授权的用户可以执行管理任务。如果授权的用户位于该系统上，则该用户将拥有 `Console User`（控制台用户）权限。

▼ 如何让用户使用自己的口令承担角色

缺省情况下，用户只有键入角色的口令才能承担角色。执行以下过程可使在 Oracle Solaris 中承担角色的过程与在 Linux 环境中承担角色的过程相似。

开始之前 您必须已承担了包括 `User Security`（用户安全）权限配置文件的角色。该角色不得是要更改其 `roleauth` 值的角色。

- 启用一个用户口令对角色进行验证。

```
$ rolemod -K roleauth=user rolename
```

为了承担该角色，指定的用户现在可以使用自己的口令，而非专为该角色创建的口令。

示例 9-26 使用权限配置文件时允许角色使用指定的用户的口令

在以下示例中，root 角色可在本地系统上更改角色 secadmin 的 roleauth 值。

```
# profiles -K roleauth=user "System Administrator"
```

当指定了 Security Administrator（安全管理员）权限配置文件的用户要承担该角色时，系统将提示用户需要输入口令。在以下序列中，角色名称为 secadmin：

```
% su - secadmin
Password:      Type user password
$             /** You are now in a profile shell with administrative rights**/
```

如果已为用户指定了其他角色，则用户也将使用自己的口令对这些角色进行验证。

示例 9-27 更改 LDAP 系统信息库中角色的 roleauth 值

在以下示例中，root 角色允许可以承担角色 secadmin 的所有用户在承担角色时使用自己的口令。对于由 LDAP 服务器管理的所有系统上的这些用户，都授予了该功能。

```
# rolemod -S ldap -K roleauth=user secadmin
# profiles -S ldap -K roleauth=user "Security Administrator"
```

故障排除 如果已对角色设置 roleauth=user，则通过用户口令，验证的角色可以访问指定给该角色的所有权限。此关键字是搜索相关关键字。有关更多信息，请参见第 187 页中的“搜索指定安全属性的顺序”。

▼ 如何将 root 角色更改为用户

当管理员淘汰已从网络中删除的系统时，可能会将 root 更改为用户。在该实例中，以 root 身份登录系统可简化清除过程。

开始之前 您必须成为分配有 User Management（用户管理）和 User Security（用户安全）权限配置文件的管理员。

- 1 从本地用户中删除 root 角色指定。
例如，从两个用户中删除角色指定。

```
% su - root
Password: a!2@3#4$5%6^7
# roles jdoe
root
# roles kdoe
root
# roles ldoe
secadmin
```

```
# usermod -R "" jdoe
# usermod -R "" kdoe
#
```

2 将 root 角色更改为用户。

```
# rolemod -K type=normal root
```

当前承担 root 角色的用户保持不变，而具有 root 访问权限的其他用户可以使用 su 命令成为 root 或以 root 用户身份登录系统。

3 验证更改。

您可以使用以下命令之一。

```
# getent user_attr root
root:::auths=solaris.*;profiles=All;audit_flags=lo\:no;lock_after_retries=no;
min_label=admin_low;clearance=admin_high
```

如果输出中缺少 type 关键字或此关键字等于 normal，则此帐户不是一个角色。

```
# userattr type root
```

如果输出为空或列出了 normal，则此帐户不是一个角色。

示例 9-28 防止 root 角色被用于配置系统

在以下示例中，站点安全策略要求应防止 root 帐户维护系统。管理员已创建和测试维护系统的角色。这些角色包括每个安全配置文件和 System Administrator（系统管理员）权限配置文件。已为一位可信用户指定了可以恢复备份的角色。没有任何角色可以更改系统、用户或权限配置文件的审计标志。

为防止使用 root 帐户维护系统，安全管理员删除了 root 的 role 指定。由于 root 帐户必须能够以单用户模式登录到系统，所以该帐户保留了一个口令。

```
# rolemod -K roles= jdoe
# userattr roles jdoe
```

示例 9-29 将 root 用户更改为 root 角色

在此示例中，root 用户将 root 用户恢复为角色。

首先，root 将 root 帐户更改为角色并验证此更改。

```
# rolemod -K type=role root
# getent user_attr root
root:::type=role;auths=solaris.*;profiles=All;audit_flags=lo\:no;
lock_after_retries=no;min_label=admin_low;clearance=admin_high
```

然后，root 将 root 角色指定给一个本地用户。

```
# usermod -R root jdoe
```

故障排除 在桌面环境中，如果 root 为角色，则您不得以 root 身份直接登录。一条诊断消息会指出 root 在您的系统中为角色。

如果您没有可以承担 root 角色的本地帐户，请创建一个。在单用户模式下以 root 身份登录到系统，创建一个本地用户帐户和口令，并将 root 角色指定给该新帐户。然后，以新用户的身份登录并承担 root 角色。

使用特权（任务）

以下任务列表指出了有关在系统上管理和使用特权的逐步说明。

任务	说明	参考
运行命令时使用特权。	涉及列出已指定给您的特权以及在系统上可用的特权。	第 175 页中的“确定您的特权（任务列表）”
在站点上使用特权。	涉及指定、删除、添加和调试特权使用。	第 179 页中的“管理特权（任务列表）”

确定您的特权（任务列表）

直接将特权指定给用户时，这些特权在每个 shell 中都有效。未直接将特权指定给用户时，用户必须打开一个配置文件 shell。例如，当具有指定特权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，用户必须在配置文件 shell 中执行此命令。

以下任务列表列出了有关查看已指定给您的特权的过程。

任务	说明	参考
查看定义的特权。	列出 Oracle Solaris 特权及其定义。	第 176 页中的“如何列出系统上的特权”
查看您作为用户在任何 shell 中的特权。	显示直接指定给您的特权。您运行的所有进程都将以这些特权运行。	第 176 页中的“如何确定已直接指定给您的特权”
查看您在某配置文件 shell 中的特权命令。	显示您可以通过指定权限配置文件运行的特权命令。	第 178 页中的“如何确定您可以运行的特权命令”
查看您作为角色在任何 shell 中的特权。	显示您的角色可以通过指定的权限配置文件运行的特权命令。	第 178 页中的“如何确定您可以运行的特权命令”

▼ 如何列出系统上的特权

以下过程显示如何查看特权名称和定义。

- 在终端窗口中，可以联机查看特权。
- 通过查看 **privileges(5)** 手册页列出特权。

```
% man privileges
Standards, Environments, and Macros          privileges(5)

NAME
    privileges - process privilege model
...
    The defined privileges are:

    PRIV_CONTRACT_EVENT

        Allow a process to request reliable delivery of events
        to an event endpoint.

        Allow a process to include events in the critical event
        set term of a template which could be generated in
        volume by the user.
...
```

此特权格式由开发者使用。

- 通过使用 **ppriv** 命令列出特权。

```
% ppriv -lv | more
contract_event
    Allows a process to request critical events without limitation.
    Allows a process to request reliable delivery of all events on
    any event queue.
...
win_upgrade_sl
    Allows a process to set the sensitivity label of a window
    resource to a sensitivity label that dominates the existing
    sensitivity label.
    This privilege is interpreted only if the system is configured
    with Trusted Extensions.
```

此特权格式用于通过使用 **useradd**、**roleadd**、**usermod** 和 **rolemod** 命令将特权指定给用户和角色，以及使用 **profiles** 命令将特权指定给权限配置文件。

▼ 如何确定已直接指定给您的特权

以下过程说明如何确定是否已直接为您指定了特权。



注意 – 不当使用直接指定的特权可能导致无意中破坏安全性。有关讨论，请参见第 137 页中的“直接指定安全属性时的安全注意事项”。

1 列出您运行的进程可以使用的特权。

有关过程，请参见第 180 页中的“如何确定进程的特权”。

2 在任何 shell 中调用操作并运行命令。

有效特权集中列出的特权在整个会话中都有效。如果已为您直接指定了除基本特权集之外的特权，则会在有效特权集中列出这些特权。

示例 9-30 确定直接指定给您的特权

如果已经为您直接指定了特权，则基本特权集包含的特权会多于缺省的基本特权集。在此示例中，用户始终能够访问 `proc_clock_highres` 特权。

```
% /usr/bin/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 9-31 确定直接指定给角色的特权

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特权。在以下示例中，已经直接将处理日期和时间程序的特权指定给角色 `realtime`。

```
% su - realtime
Password: <Type realtime password>
$ /usr/bin/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
```

▼ 如何确定您可以运行的特权命令

未直接将特权指定给用户时，此用户可通过权限配置文件获取对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。

1 确定已指定给您的权限配置文件。

```
% profiles
Audit Review
Console User
Suspend To RAM
Suspend To Disk
Brightness
CPU Power Management
Network Autoconf
Desktop Print Management
Network Wifi Info
Desktop Removable Media User
Basic Solaris User
All
```

2 确定您的 Audit Review（审计查看）配置文件权限。

```
profiles -l
Audit Review

solaris.audit.read

/usr/sbin/auditreduce  euid=0
/usr/sbin/auditstat    euid=0
/usr/sbin/praudit      euid=0
```

通过 Audit Review（审计查看）权限配置文件，您可以使用有效 UID 0 运行 `auditreduce`、`auditstat` 和 `praudit` 命令，并为您指定 `solaris.audit.read` 授权。

示例 9-32 确定角色的特权命令

在以下示例中，用户将承担指定的角色并列出其中一个权限配置文件中包括的命令。

```
% roles
devadmin
% su - devadmin
Password:      Type devadmin password
$ profiles -l
Device Security
    /usr/bin/kbd          uid=0;gid=sys
    /usr/sbin/add_allocatable  euid=0
    /usr/sbin/add_drv        uid=0
    /usr/sbin/devfsadm        uid=0
    /usr/sbin/eeprom          uid=0
    /usr/sbin/list_devices    euid=0
    /usr/sbin/rem_drv          uid=0
    /usr/sbin/remove_allocatable  euid=0
```

```
/usr/sbin/strace      euid=0
/usr/sbin/update_drv   uid=0
```

示例 9-33 运行您的角色中的特权命令

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:      <Type admin password>
$ /usr/bin/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
```

管理特权（任务列表）

管理用户和角色特权的最安全的方法是将特权使用限制在权限配置文件内的命令中。然后，在角色中包括该权限配置文件。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。以下过程说明了如何指定特权、删除特权以及调试特权使用。

以下任务列表列出了有关指定、删除和调试特权以及运行包含特权命令的脚本的过程。

任务	说明	参考
确定进程中的特权。	列出进程的有效特权集、可继承特权集、允许特权集和限制特权集。	第 180 页中的“如何确定进程的特权”
确定进程中缺少的特权。	列出成功运行失败进程所需的特权。	第 181 页中的“如何确定程序所需的特权”
为命令添加特权。	在权限配置文件中为命令添加特权。可以为用户或角色指定权限配置文件。然后，用户可以在配置文件 shell 中运行具有指定特权的命令。	示例 9-14
为用户或角色指定特权。	扩展用户或角色的可继承特权集。使用此过程时应谨慎。	示例 9-24
限制用户的特权。	限制用户的基本特权集。使用此过程时应谨慎。	示例 9-12

任务	说明	参考
运行特权 shell 脚本。	为 shell 脚本和 shell 脚本中的命令添加特权。然后，在配置文件 shell 中运行此脚本。	第 183 页中的“如何运行具有特权命令的 Shell 脚本”

▼ 如何确定进程的特权

此过程说明如何确定可用于进程的特权。列出内容不包括已经指定给特定命令的特权。

● 列出可用于您的 shell 进程的特权。

```
% ppriv pid
$ ppriv -v pid

pid    进程号。使用双美元符号 ($$) 将父 shell 的进程号传递给命令。
-v     提供特权名称的详细列表。
```

示例 9-34 确定当前 shell 中的特权

在以下示例中，列出了用户 shell 进程的父进程中的特权。在下面第二个示例中，列出了特权的全名。输出中的单个字母指代以下特权集：

- E 有效特权集。
- I 可继承特权集。
- P 允许特权集。
- L 限制特权集。

```
% ppriv $$
1200:  -csh
flags = <none>
      E: basic
      I: basic
      P: basic
      L: all

% ppriv -v $$
1200:  -csh
flags = <none>
      E: file_link_any,net_access,proc_exec,proc_fork,proc_info,proc_session
      I: file_link_any,net_access,proc_exec,proc_fork,proc_info,proc_session
      P: file_link_any,net_access,proc_exec,proc_fork,proc_info,proc_session
      L: cpc_cpu,dtrace_kernel,dtrace_proc,dtrace_user,...,sys_time
```

示例 9-35 确定可承担的角色角色的特权

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特权。在以下示例中，角色 sysadmin 不具有直接指定的特权。


```
% su - sysadmin
Password: <Type sysadmin password>
$ /usr/bin/whoami
sysadmin
$ ppriv -v $$
1400: pfksh
flags = <none>
E: file_link_any,file_read,file_write,net_access,proc_exec,proc_fork,
   proc_info,proc_session
I: file_link_any,file_read,file_write,net_access,proc_exec,proc_fork,
   proc_info,proc_session
P: file_link_any,file_read,file_write,net_access,proc_exec,proc_fork,
   proc_info,proc_session
L: cpc_cpu,dtrace_kernel,dtrace_proc,dtrace_user,...,win_upgrade_sl
```

▼ 如何确定程序所需的特权

此过程可确定成功运行命令或进程所需的特权。

开始之前 此调试过程必须在命令或进程失败后才有效。

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[5245]: missing privilege "file_dac_write"
          (euid = 130, syscall = 224) needed at zfs_zaccess+0x258
touch: cannot create /etc/acct/yearly: Permission denied
```

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 9-36 使用 truss 命令检查特权使用

truss 命令可以在常规 shell 中调试特权使用。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特权。

示例 9-37 使用 ppriv 命令在配置文件 Shell 中检查特权使用

ppriv 命令可以在配置文件 shell 中调试特权使用。如果为用户指定了权限配置文件，并且此权限配置文件包括具有特权的命令，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，这些命令执行时不会使用特权。

在此示例中，jdoe 用户可以承担角色 objadmin。objadmin 角色拥有 Object Access Management（对象访问管理）权限配置文件。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 alooe staff 2303 Apr 10 10:10 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 zfs_zaccess+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: <Type objadmin password>
$ ls -l useful.script
-rw-r--r-- 1 alooe staff 2303 Apr 10 10:10 useful.script
$ chown objadmin useful.script
$ ls -l useful.script
-rw-r--r-- 1 objadmin staff 2303 Apr 10 10:10 useful.script
$ chgrp admin useful.script
$ ls -l objadmin.script
-rw-r--r-- 1 objadmin admin 2303 Apr 10 10:11 useful.script
```

示例 9-38 更改 root 用户拥有的文件

此示例说明了防止特权升级的方法。有关讨论，请参见第 195 页中的“防止特权升级”。此文件归 root 用户所有。由于权限较低的 objadmin 角色需要所有特权才能更改文件的所有权，因此操作失败。

```
jdoe% su - objadmin
Password: <Type objadmin password>
$ cd /etc; ls -l system
-rw-r--r-- 1 root sys 1883 Oct 10 10:20 system
$ chown objadmin system
chown: system: Not owner
$ ppriv -eD chown objadmin system
chown[11481]: missing privilege "ALL"
```

```
(euid = 101, syscall = 16) needed at zfs_zaccess+0x258
chown: system: Not owner
```

▼ 如何运行具有特权命令的 Shell 脚本

注 - 创建运行需要特权的命令的 shell 脚本时，相应的权限配置文件必须包含具有指定特权的命令。

开始之前 您必须是 root 角色。

- 1 将 `/bin/pfsh` 或任何其他配置文件 `shell` 放在脚本的第一行。

```
#!/bin/pfsh
# Copyright (c) 2011 by Oracle
```

- 2 确定脚本中的命令所需的特权。

```
% ppriv -eD script-full-path
```

- 3 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

- 4 创建或修改脚本的权限配置文件。

您需要将 shell 脚本、该脚本中的命令以及所需的安全属性添加到权限配置文件中。有关步骤，请参见第 159 页中的“如何创建或更改权限配置文件”。

- 5 将权限配置文件添加到某个角色中，并将此角色指定给用户。

为了运行脚本，用户将承担该角色，并在该角色的配置文件 `shell` 中运行脚本。

- 要向角色添加权限配置文件，请参见第 167 页中的“如何更改角色的安全属性”。
- 要将角色指定给用户，请参见示例 9-20。

Oracle Solaris 中的安全属性（参考）

本章提供了有关 RBAC 和特权的参考资料。以下是本章中参考信息的列表：

- 第 185 页中的“权限配置文件”
- 第 187 页中的“搜索指定安全属性的顺序”
- 第 188 页中的“Authorizations（授权）”
- 第 189 页中的“RBAC 数据库”
- 第 192 页中的“RBAC 命令”
- 第 194 页中的“用于处理特权的管理命令”
- 第 194 页中的“包含特权信息的文件”
- 第 195 页中的“特权和审计”
- 第 195 页中的“防止特权升级”
- 第 196 页中的“传统应用程序和特权模型”

有关使用 RBAC 的信息，请参见第 9 章，使用基于角色的访问控制（任务）。有关概述信息，请参见第 127 页中的“基于角色的访问控制（概述）”。

要使用特权，请参见第 175 页中的“使用特权（任务）”。有关概述信息，请参见第 138 页中的“特权（概述）”。

权限配置文件

本节介绍了一些典型的权限配置文件。权限配置文件是以下对象的便利集合：授权和其他安全属性、具有安全属性的命令以及补充权限配置文件。Oracle Solaris 提供了多种权限配置文件。如果这些文件无法满足您的需要，您可以修改现有权限配置文件，创建新的权限配置文件。

权限配置文件必须按权限从高到低的顺序指定。有关更多信息，请参见第 187 页中的“搜索指定安全属性的顺序”。

- **System Administrator（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可用于创建功能强大的角色的其他配置文件。请注意，将在补充权限配置文件列表的末尾指定 All（所有）权限配置文件。profiles 命令显示配置文件的内容。

```
% profiles -p "System Administrator" info
```

- **Operator（操作员）权限配置文件**—提供可用于管理文件和脱机介质的有限功能。此配置文件包括可用于创建简单角色的补充权限配置文件。profiles 命令显示配置文件的内容。

```
% profiles -p Operator info
```

- **Printer Management（打印机管理）权限配置文件**—提供用于处理打印的数量有限的命令和授权。此配置文件是涉及单个管理区域的若干个配置文件之一。profiles 命令显示配置文件的内容。

```
% profiles -p "Printer Management" info
```

- **Basic Solaris User（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。请注意，Basic Solaris User（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。需要更严格的安全性的站点可能更倾向于从 policy.conf 文件中删除此配置文件，或者指定 Stop（停止）权限配置文件。profiles 命令显示配置文件的内容。

```
% profiles -p "Basic Solaris User" info
```

- **Console User（控制台用户）权限配置文件**—为工作站所有者提供对计算机前用户的授权、命令和操作的访问权限。profiles 命令显示配置文件的内容。

```
% profiles -p "Console User" info
```

- **All（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。此配置文件适用于具有有限权限的用户。profiles 命令显示配置文件的内容。

```
% profiles -p All info
```

- **Stop（停止）权限配置文件**—是一个特殊的权限配置文件，可停止对其他权限配置文件的评估。此配置文件可阻止对 policy.conf 文件中的 AUTHS_GRANTED、PROFS_GRANTED 和 CONSOLE_USER 变量的评估。通过此配置文件，可为角色和用户提供受限制的配置文件 shell。

注—Stop（停止）权限配置文件间接影响特权指定。列在 Stop（停止）配置文件之后的权限配置文件不会被评估。因此，使用这些配置文件中特权的命令不会生效。要使用此配置文件，请参见第 171 页中的[“如何将管理员限于显式指定的权限”](#)。

profiles 命令显示配置文件的内容。

```
% profiles -p Stop info
```

每个权限配置文件都有关联的帮助文件。这些帮助文件以 HTML 格式提供，并且可定制。这些文件位于 `/usr/lib/help/profiles/locale/C` 目录下。

查看权限配置文件的内容

可以通过三种方式查看权限配置文件的内容。

- 使用 `getent` 命令，可以查看系统中所有权限配置文件的内容。有关示例输出，请参见第 146 页中的“如何查看所有已定义的安全属性”。
- 使用 `profiles -p "Profile Name" info` 命令，可以查看特定权限配置文件的内容。
- 使用 `profiles -l account-name` 命令，可以查看指定给特定用户或角色的权限配置文件的内容。

有关更多信息，请参见 `getent(1M)` 和 `profiles(1)` 手册页。

搜索指定安全属性的顺序

可以直接或通过权限配置文件为用户或角色指定安全属性。搜索顺序影响使用的安全属性值。将使用查找到的第一个属性实例的值。

注 - 授权的顺序不重要。授权是累积的。

用户登录时，按以下搜索顺序指定安全属性：

- 使用 `useradd` 和 `usermod` 命令指给用户的**安全属性**。有关列表，请参见第 189 页中的“`user_attr` 数据库”。
- 使用 `useradd` 和 `usermod` 命令指给用户的**权限配置文件**。按顺序对这些指定进行搜索。
 首先是列表中的第一个配置文件，其权限配置文件列表；然后是列表中的第二个配置文件，其配置文件列表，等等。系统使用第一个实例的值，但 `auths` 值除外，该值是累积的。权限配置文件中的属性包含用户的所有安全属性，以及补充的配置文件。有关列表，请参见第 189 页中的“`user_attr` 数据库”。
- **Console User（控制台用户）权限配置文件值**。有关说明，请参见第 185 页中的“**权限配置文件**”。
- 如果指定了 **Stop（停止）权限配置文件**，将停止对安全属性的评估。指定了 **Stop（停止）配置文件** 之后不再指定任何属性。**Stop（停止）配置文件** 在 **Console User（控制台用户）权限配置文件** 之后、`policy.conf` 文件中的其他安全属性（包括 `AUTHS_GRANTED`）之前进行评估。有关说明，请参见第 185 页中的“**权限配置文件**”。
- `policy.conf` 文件中的 **Basic Solaris User（基本 Solaris 用户）权限配置文件值**。

- `policy.conf` 文件中的 **AUTHS_GRANTED** 值。
- `policy.conf` 文件中的 **PROFS_GRANTED** 值。
- `policy.conf` 文件中的 **PRIV_DEFAULT** 值。
- `policy.conf` 文件中的 **PRIV_LIMIT** 值。

Authorizations (授权)

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，将通过遵循 RBAC 的应用程序检查授权。

授权是用户级的，因此可以扩展。可以编写要求授权的程序，为系统添加授权，为这些授权创建权限配置文件，并将该权限配置文件指给允许使用该程序的用户或角色。

授权命名约定

授权具有内部使用的名称。例如，`solaris.system.date` 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，`Set Date & Time` 是 `solaris.system.date` 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、任何子区域以及功能组成。授权名称的各个部分以点分隔。例如，`com.xyzcorp.device.access` 便是一个授权名称。此约定的例外是 Oracle Solaris 的授权，它使用前缀 `solaris` 而不是 Internet 名称。使用命名约定，管理员可以用层次化方式应用授权。通配符 (*) 可以表示点右侧的所有字符串。

授权粒度示例

作为授权使用方法的示例，考虑以下情况：处于 Network Link Security 角色的用户受限于 `solaris.network.link.security` 授权，而 Network Security 角色以 Network Link Security（网络链路安全）权限配置文件作为补充的配置文件，另加 `solaris.network.*` 和 `solaris.smf.manage.ssh` 授权。

授权中的委托授权

使用以后缀 `delegate` 结尾的授权，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。

使用 `solaris.auth.delegate` 授权，用户或角色可将指给他们的任何授权委托给其他用户。

例如，拥有 `solaris.auth.delegate` 和 `solaris.network.wifi.wep` 授权的角色可将 `solaris.network.wifi.wep` 授权委托给其他用户或角色。类似地，拥有 `solaris.auth.delegate` 和 `solaris.network.wifi.wep` 授权的角色可将 `solaris.network.wifi.wep` 授权委托给其他用户或角色。

RBAC 数据库

以下数据库存储 RBAC 元素的数据：

- **扩展用户属性数据库** (`user_attr`) — 将用户和角色与授权、特权、关键字和权限配置文件关联
- **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、特权、关键字，以及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具有安全属性的命令

`policy.conf` 数据库包含应用于所有用户的授权、特权和权限配置文件。有关更多信息，请参见第 191 页中的“[policy.conf 文件](#)”。

RBAC 数据库和命名服务

在 SMF 服务中，针对名称服务转换参数 `svc:/system/name-service/switch` 定义 RBAC 数据库的名称服务范围。此服务中针对 RBAC 数据库的属性为 `auth_attr`、`password` 和 `prof_attr`。`password` 属性设置 `passwd` 和 `user_attr` 数据库的命名服务优先级。`prof_attr` 属性设置 `prof_attr` 和 `exec_attr` 数据库的命名服务优先级。

在以下输出中，`auth_attr`、`password` 和 `prof_attr` 条目未列出。因此，RBAC 数据库使用的是 `files` 命名服务。

```
# svccfg -s name-service/switch listprop config
config                                application
config/value_authorization          astring      solaris.smf.value.name-service.switch
config/default                      astring      files
config/host                         astring      "files ldap dns"
config/printer                      astring      "user files ldap"
```

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。

以下安全属性可使用 `roleadd`、`rolemod`、`useradd`、`usermod` 和 `profiles` 命令进行设置：

- 对于用户，`roles` 关键字指定一个或多个已定义的角色。
- 对于角色，使用 `roleauth` 关键字的 `user` 值可允许角色使用用户口令而不是角色口令进行授权。缺省情况下，值为 `role`。
- 对于用户或角色，可设置以下属性：
 - `audit_flags` 关键字—修改审计掩码。有关参考信息，请参见 [audit_flags\(5\)](#) 手册页。
 - `auths` 关键字—指定授权。有关参考信息，请参见 [auths\(1\)](#) 手册页。
 - `defaultpriv` 关键字—添加特权或者从缺省基本特权集中删除特权。有关参考信息，请参见第 141 页中的“如何实现特权”。
 - `limitpriv` 关键字—添加特权或者从缺省限制特权集中删除特权。有关参考信息，请参见第 141 页中的“如何实现特权”。

这些特权始终有效，它们不是命令的属性。有关参考信息，请参见 [privileges\(5\)](#) 手册页和第 141 页中的“如何实现特权”。

 - `projects` 关键字—添加缺省项目。有关参考信息，请参见 [project\(4\)](#) 手册页。
 - `lock_after_retries` 关键字—如果值为 `yes`，当重试次数超出 `/etc/default/login` 文件中允许的次數后，系统会锁定。
 - `profiles` 关键字—指定权限配置文件。

有关更多信息，请参见 [user_attr\(4\)](#) 手册页。要查看此数据库的内容，请使用 `getent user_attr` 命令。有关更多信息，请参见 [getent\(1M\)](#) 手册页和第 146 页中的“如何查看所有已定义的安全属性”。

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。

要查看此数据库的内容，请使用 `getent prof_attr` 命令。有关更多信息，请参见 [getent\(1M\)](#) 手册页和第 146 页中的“如何查看所有已定义的安全属性”。

prof_attr 数据库

`prof_attr` 数据库存储指定给权限配置文件的名称、说明、帮助文件位置、特权以及授权。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 191 页中的“`exec_attr` 数据库”。

有关更多信息，请参见 [prof_attr\(4\)](#) 手册页。要查看此数据库的内容，请使用 `getent exec_attr` 命令。有关更多信息，请参见 [getent\(1M\)](#) 手册页和第 146 页中的“[如何查看所有已定义的安全属性](#)”。

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或用户运行。

有关更多信息，请参见 [exec_attr\(4\)](#) 手册页。要查看此数据库的内容，请使用 `getent` 命令。有关更多信息，请参见 [getent\(1M\)](#) 手册页和第 146 页中的“[如何查看所有已定义的安全属性](#)”。

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、特定授权和特定特权的方法。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。
- `CONSOLE_USER=Console User`—指 Console User（控制台用户）权限配置文件。此配置文件附带了一组便利的控制台用户授权。可以定制此配置文件。要查看此配置文件的内容，请参见第 185 页中的“[权限配置文件](#)”。
- `PRIV_DEFAULT=privileges`—指一个或多个特权。
- `PRIV_LIMIT=privileges`—指所有特权。

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
```

有关特权的更多信息，请参见第 138 页中的“[特权（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

以下命令检索和设置 RBAC 信息。

表 10-1 RBAC 管理命令

命令的手册页	说明
auths(1)	显示用户的授权。
getent(1M)	用于列出 <code>user_attr</code> 、 <code>prof_attr</code> 和 <code>exec_attr</code> 数据库内容的接口。
nscd(1M)	名称服务高速缓存守护进程，适用于高速缓存 <code>user_attr</code> 、 <code>prof_attr</code> 和 <code>exec_attr</code> 数据库。使用 <code>svcadm</code> 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。
pfexec(1)	由配置文件 <code>shell</code> 使用以执行在 <code>exec_attr</code> 数据库中指定的具有安全属性的命令。
policy.conf(4)	系统安全策略的配置文件。列出授予的授权、授予的特权和其他安全信息。
profiles(1)	显示某个指定用户的权限配置文件。创建或修改本地系统或 LDAP 网络中的权限配置文件。
roles(1)	显示指定用户可以承担的角色。
roleadd(1M)	向本地系统或 LDAP 网络中添加角色。
roleadd(1M)	向本地系统或 LDAP 网络中添加角色。
rolemod(1M)	修改本地系统或 LDAP 网络中角色的属性。
userattr(1)	显示指定给用户或角色帐户的特定权限的值。
useradd(1M)	向系统或 LDAP 网络中添加用户帐户。-R 选项将角色指定给用户帐户。
userdel(1M)	从系统或 LDAP 网络中删除用户登录。
usermod(1M)	修改系统上的用户帐户属性。

要求授权的选定命令

下表提供了在 Oracle Solaris 系统上如何使用授权限制命令选项的示例。有关授权的更多讨论，请参见第 188 页中的“[Authorizations（授权）](#)”。

表 10-2 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 <code>solaris.jobs.user</code> (<code>at.allow</code> 和 <code>at.deny</code> 文件都不存在时)
atq(1)	所有选项所需的 <code>solaris.jobs.admin</code>
cdrw(1)	所有选项所需的 <code>solaris.device.cdrw</code> ，缺省情况下在 <code>policy.conf</code> 文件中授予
crontab(1)	选项提交作业所需的 <code>solaris.jobs.user</code> (<code>crontab.allow</code> 和 <code>crontab.deny</code> 文件都不存在时) 选项列出或修改其他用户的 <code>crontab</code> 文件所需的 <code>solaris.jobs.admin</code>
allocate(1)	分配设备所需的 <code>solaris.device.allocate</code> (或在 <code>device_allocate</code> 文件中指定的其他授权) 将设备分配给其他用户 (<code>-F</code> 选项) 所需的 <code>solaris.device.revoke</code> (或在 <code>device_allocate</code> 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 <code>solaris.device.allocate</code> (或在 <code>device_allocate</code> 文件中指定的其他授权) 强制解除指定设备的分配 (<code>-F</code> 选项) 或所有设备的分配 (<code>-I</code> 选项) 所需的 <code>solaris.device.revoke</code> (或在 <code>device_allocate</code> 中指定的其他授权)
list_devices(1)	列出其他用户的设备 (<code>-U</code> 选项) 所需的 <code>solaris.device.revoke</code>
roleadd(1M)	创建角色需要 <code>solaris.user.manage</code> 。设置初始口令需要 <code>solaris.account.activate</code> 。设置口令策略 (如帐户锁定和口令生命期) 需要 <code>solaris.account.setpolicy</code> 。
roledel(1M)	删除口令需要 <code>solaris.passwd.assign</code> 授权。
rolemod(1M)	更改口令需要 <code>solaris.passwd.assign</code> 授权。更改口令策略 (如帐户锁定和口令生命期) 需要 <code>solaris.account.setpolicy</code> 。
sendmail(1M)	访问邮件子系统功能所需的 <code>solaris.mail</code> ；查看邮件队列所需的 <code>solaris.mail.mailq</code>
useradd(1M)	创建用户需要 <code>solaris.user.manage</code> 。设置初始口令需要 <code>solaris.account.activate</code> 。设置口令策略 (如帐户锁定和口令生命期) 需要 <code>solaris.account.setpolicy</code> 。
userdel(1M)	删除口令需要 <code>solaris.passwd.assign</code> 授权。
usermod(1M)	更改口令需要 <code>solaris.passwd.assign</code> 授权。更改口令策略 (如帐户锁定和口令生命期) 需要 <code>solaris.account.setpolicy</code> 。

特权

特权限制进程是在内核中实现的，可以在命令、用户、角色和系统级别对进程加以限制。

用于处理特权的管理命令

下表列出了可用于处理特权的命令。

表 10-3 用于处理特权的命令

目的	命令	手册页
检查进程特权	<code>ppriv -v pid</code>	ppriv(1)
设置进程特权	<code>ppriv -s spec</code>	
列出系统上的特权	<code>ppriv -l</code>	
列出特权及其说明	<code>ppriv -lv priv</code>	
调试特权问题	<code>ppriv -eD failed-operation</code>	
为新用户指定特权	<code>useradd</code>	useradd(1M)
为现有用户添加特权	<code>usermod</code>	usermod(1M)
为权限配置文件指定特权	<code>profiles</code>	profiles(1)
为新角色指定特权	<code>roleadd</code>	roleadd(1M)
为现有角色添加特权	<code>rolemod</code>	rolemod(1M)
查看设备策略	<code>getdevpolicy</code>	getdevpolicy(1M)
设置设备策略	<code>devfsadm</code>	devfsadm(1M)
更新关于打开的设备的设备策略	<code>update_drv -p policy driver</code>	update_drv(1M)
为设备添加设备策略	<code>add_drv -p policy driver</code>	add_drv(1M)

包含特权信息的文件

以下文件包含有关特权的信息。

表 10-4 包含特权信息的文件

文件和手册页	特权信息	说明
/etc/security/policy.conf	PRIV_DEFAULT	系统的可继承特权集
policy.conf(4)	PRIV_LIMIT	系统的限制特权集
syslog.conf	调试消息的系统日志文件	特权调试日志
syslog.conf(4)	priv.debug 项中设置的路径	

特权和审计

可以对特权的使用进行审计。只要进程使用特权，就会在 `upriv` 审计标记的审计迹中记录该特权的使用。如果记录中包含特权名称，则将使用特权名称的文本表示形式。以下审计事件记录特权的使用：

- **AUE_SETPPRIV 审计事件**—更改了特权集时，该事件将生成一条审计记录。AUE_SETPPRIV 审计事件在 `pm` 类中。
- **AUE_MODALLOCPRIV 审计事件**—从内核外部添加特权时，该审计事件将生成一条审计记录。AUE_MODALLOCPRIV 审计事件在 `ad` 类中。
- **AUE_MODDEVPLCY 审计事件**—更改了设备策略时，该审计事件将生成一条审计记录。AUE_MODDEVPLCY 审计事件在 `ad` 类中。
- **AUE_PFEXEC 审计事件**—如果在启用 `pfexec()` 时调用 `execve()`，审计事件将生成一条审计记录。AUE_PFEXEC 审计事件在 `as`、`ex`、`ps` 和 `ua` 审计类中。特权的名称包含在审计记录中。

不会审计基本特权集中特权的成功使用。但是，将会审计对已从用户基本特权集中删除的基本特权的使用。

防止特权升级

内核可防止**特权升级**。特权升级是指某特权允许进程执行超出其应有功能的功能。为防止进程获取超出其应有特权的特权，对易受攻击的系统进行修改需要具有完全特权集。例如，只有具有完全特权集的进程才能更改 `root (UID=0)` 拥有的文件或进程。`root` 帐户不需要具有特权就能更改 `root` 拥有的文件。但是，非超级用户必须具有全部特权才能更改 `root` 拥有的文件。

类似地，提供设备访问的操作需要具有有效特权集中的所有特权。

`file_chown_self` 和 `proc_owner` 特权可进行特权升级。`file_chown_self` 特权允许进程放弃其文件。`proc_owner` 特权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` 特权。如果将 `rstchown` 变量设置为零，则将从系统和所有用户的初始可继承特权集中删除 `file_chown_self` 特权。有关 `rstchown` 系统变量的更多信息，请参见 [chown\(1\)](#) 手册页。

`file_chown_self` 特权非常安全地指定给特定命令，放置在配置文件中，并指定给角色供配置文件 `shell` 使用。

`proc_owner` 特权不足以将进程 `UID` 切换为 `0`。要将进程从任何 `UID` 切换为 `UID=0` 需要具有全部特权。由于 `proc_owner` 特权授予对系统上所有文件的无限制读取权限，因此可以非常安全地将该特权指定给特定命令，放置在配置文件中，并指定给角色供配置文件 `shell` 使用。



注意 – 可以修改用户帐户，将 `file_chown_self` 特权或 `proc_owner` 特权包括在该用户的初始可继承特权集中。要将这些强大的特权放置在任何用户、角色或系统的可继承特权集中，您应有充分的安全理由。

有关如何防止设备特权升级的详细信息，请参见第 144 页中的“特权和设备”。

传统应用程序和特权模型

为了适应传统应用程序，特权的实现使用超级用户模型和特权模型。内核会自动跟踪 `PRIV_AWARE` 标志，此标志指示某个程序设计为可使用特权。请考虑不能识别特权的子进程。从父进程继承的所有特权均可在子进程的允许特权集和有效特权集中找到。如果子进程将 `UID` 设置为 `0`，则其可能不具有完全超级用户功能。进程的有效特权集和允许特权集会被限制为子进程限制特权集中包含的特权。这样，可识别特权的进程的限制特权集会限制不能识别特权的子进程的超级用户特权。

第 4 部分

加密服务

本节介绍 Oracle Solaris 提供的集中加密和公钥技术功能。

- 第 11 章，加密框架（概述）
- 第 12 章，加密框架（任务）
- 第 13 章，密钥管理框架

加密框架（概述）

本章介绍 Oracle Solaris 的加密框架功能。以下是本章中信息的列表。

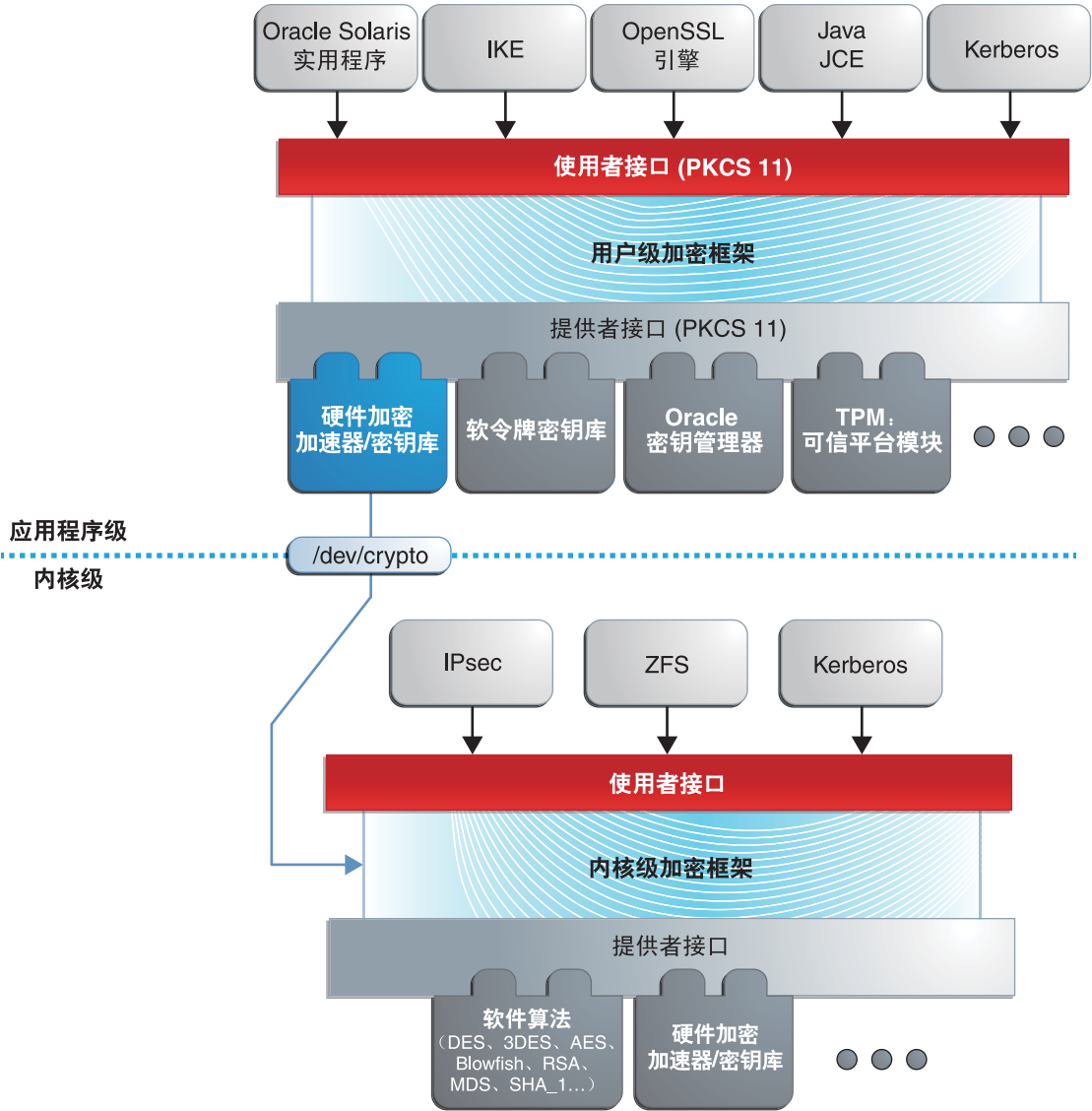
- 第 199 页中的“加密框架简介”
- 第 201 页中的“加密框架术语”
- 第 202 页中的“加密框架的范围”
- 第 203 页中的“加密框架中的管理命令”
- 第 203 页中的“加密框架中的用户级命令”
- 第 204 页中的“加密框架的插件”
- 第 204 页中的“加密服务和区域”

要管理和使用加密框架，请参见第 12 章，加密框架（任务）。

加密框架简介

加密框架提供了一个包含算法和 PKCS #11 库的公共存储区来处理加密要求。PKCS #11 库依据以下标准实现：RSA Security Inc. 的 PKCS #11 加密令牌接口 (Cryptographic Token Interface, Cryptoki)。

图 11-1 加密框架级别



在内核级别，该框架目前可处理 Kerberos 和 IPsec 加密要求。用户级使用者包括 `libsasl` 和 `IKE`。内核 SSL (`kssl`) 代理使用加密框架。有关更多信息，请参见《Oracle Solaris 管理：网络服务》中的“使用安全套接字层协议的 Web 服务器”和 `ksslcfg(1M)` 手册页。

美国出口法要求应限制使用开放式加密接口。加密框架通过要求对内核加密服务提供者和 PKCS #11 加密服务提供者进行签名来满足现行法律规定。有关进一步介绍，请参见第 203 页中的“[第三方软件的二进制文件签名](#)”。

该框架允许加密服务提供者让 Oracle Solaris 中的许多使用者使用其服务。提供者的另一个名称是**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`）提供服务的共享目标文件。
- **内核级插件**—在软件中提供加密算法（如 [AES](#)）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。•Niagara 芯片、ncp 和 n2cp 设备驱动程序就是其示例。硬件加速器承担了操作系统的高开销加密功能负载。Sun Crypto Accelerator 6000 板是一个示例。

框架为用户级提供者实现了标准接口 PKCS #11 v2.11 库。第三方应用程序可以使用该库来访问提供者。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 实用程序在安装第三方软件时将添加这些插件。有关说明框架主要组件的图示，请参见《[Oracle Solaris 11 开发者安全性指南](#)》中的第 8 章“[Oracle Solaris 加密框架介绍](#)”。

加密框架术语

以下定义和示例列表在使用加密框架时非常有用。

- **算法**—加密算法。这些算法是已确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同一个密钥。非对称算法用于公钥加密中，要求使用两个不同的密钥。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
- **使用者**—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。

使用者示例包括：

- 应用程序，如 IKE
- 最终用户，如运行 `encrypt` 命令的一般用户。
- 内核操作，例如 IPsec
- **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**—提供框架中装入的其他插槽的功能组合的单个插槽。Metaslot 简化了处理可通过框架使用的提供者全部功能的工作。使用 metaslot 的应用程序请求某个操作时，metaslot 将确定应执行该操作的实际插槽。可以配置 metaslot 功能，但这不是必须的。缺省情况下，metaslot 处于打开状态。有关配置 metaslot 的信息，请参见 [cryptoadm\(1M\)](#) 手册页。
- **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）是不同的模式。AES 算法有两种模式：CKM_AES_ECB 和 CKM_AES_CBC。
- **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，所有提供者和所有机制都可用。禁用任何机制是对策略的应用。启用已禁用的机制也是对策略的应用。
- **提供者**—使用者使用的加密服务。提供者插入到框架中，因此也称为**插件**。

提供者示例包括：

- 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 Sun Crypto Accelerator 6000 的 mca 驱动程序
- **插槽**—到一个或多个加密设备的接口。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。
- **令牌**—在插槽中，令牌提供框架中加密设备的逻辑视图。

加密框架的范围

框架为管理员、用户和开发提供者的开发者提供命令：

- **管理命令**—cryptoadm 命令提供了 list 子命令，用于列出可用提供者及其功能。一般用户可以运行 cryptoadm list 和 cryptoadm --help 命令。

所有其他 cryptoadm 子命令要求您承担拥有 Crypto Management（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 kcfld 守护进程和在内核中刷新加密策略。有关更多信息，请参见 [svcadm\(1M\)](#) 手册页。

- **用户级命令**—digest 和 mac 命令可提供文件完整性服务。encrypt 和 decrypt 命令，用于保护文件不被窃听。有关使用这些命令的信息，请参见第 205 页中的“[使用加密框架保护文件（任务列表）](#)”。

加密框架中的管理命令

`cryptoadm` 命令可管理正在运行的加密框架。该命令包括在 `Crypto Management`（加密管理）权限配置文件中。可以将此配置文件指定给角色，以实现加密框架的安全管理。`cryptoadm` 命令可管理以下任务：

- 显示加密提供者的信息
- 禁用或启用提供者机制
- 禁用或启用 `metaslot`

`svcadm` 命令用于启用、刷新和禁用加密服务守护进程 `kcfcd`。此命令是 Oracle Solaris 的服务管理工具 (Service Management Facility, 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。

加密框架中的用户级命令

加密框架提供了用户级命令，用于检查文件完整性、加密和解密文件。独立命令 `elfsign` 使提供者可以对要在框架中使用的二进制文件进行签名。

- **digest 命令**—计算一个或多个文件或 `stdin` 的 [message digest](#)（消息摘要）。摘要对检验文件的完整性非常有用。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](#)（消息验证码）。MAC 可将数据与通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。使用 `sha1_mac` 和 `md5_hmac` 机制可以计算 MAC。
- **encrypt 命令**—使用对称密码算法加密文件或 `stdin`。`encrypt -l` 命令可以列出可用的算法。用户级库下列出的机制可用于 `encrypt` 命令。框架提供了 AES、DES、3DES（Triple-DES，三重 DES）和 ARCFOUR 机制来实现用户加密。
- **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加密框架的提供者进行签名的方法。通常，此命令由提供者的开发者运行。

`elfsign` 命令包含一些用于申请证书、对二进制文件进行签名以及验证二进制文件上签名的子命令。加密框架无法使用未签名的二进制文件。具有可验证已签名二进制文件的提供者可以使用框架。

加密框架的插件

第三方可以将其提供者插入到加密框架中。第三方提供者可以是以下对象之一：

- PKCS #11 共享库
- 可装入内核软件模块，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

提供者目标文件必须由 Oracle 提供的证书签名。证书申请基于第三方选择的私钥，以及 Oracle 提供的证书。证书申请将会发送到 Oracle，Oracle 注册第三方然后颁发该证书。之后第三方使用 Oracle 提供的证书对其提供者目标文件进行签名。

可装入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。注册通过加密框架 SPI（service provider interface，服务提供者接口）进行。

加密服务和区域

全局区域和每个非全局区域都有其自己的 `/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，并且设置此全局区域的用户级策略和系统的内核策略。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，并且设置该区域的用户级策略。内核策略由全局区域设置。

有关区域的更多信息，请参见《[Oracle Solaris 管理：Oracle Solaris Zones、Oracle Solaris 10 Zones 和资源管理](#)》中的第 II 部分,“Oracle Solaris Zones”。有关用于管理持久性应用程序的 SMF 的更多信息，请参见《[Oracle Solaris 管理：常见任务](#)》中的第 6 章“管理服务（概述）”和 `smf(5)` 手册页。

加密框架（任务）

本章介绍如何使用加密框架。以下是本章中信息的列表。

- 第 205 页中的“使用加密框架（任务列表）”
- 第 205 页中的“使用加密框架保护文件（任务）”
- 第 218 页中的“管理加密框架（任务）”

使用加密框架（任务列表）

以下任务列表列出了有关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。阻止入侵者读取文件。这些过程可以由一般用户完成。	第 205 页中的“使用加密框架保护文件（任务列表）”
管理框架。	添加、配置和删除软件提供者。禁用和启用硬件提供者机制。这些过程属于管理工作。	第 218 页中的“管理加密框架（任务列表）”

使用加密框架保护文件（任务）

本节介绍如何生成对称密钥、如何创建校验和来检验文件完整性以及如何避免文件遭到窃听。本节中的命令可由一般用户运行。开发者可以编写使用这些命令的脚本。

使用加密框架保护文件（任务列表）

加密框架可帮助您保护文件。以下任务列表列出了用于列出可用算法以及通过加密方式保护文件的过程。

任务	说明	参考
生成对称密钥。	生成随机密钥，以用于用户指定的算法。	第 206 页中的“如何使用 dd 命令生成对称密钥”
	生成具有用户指定长度的密钥。或者，可以选择将密钥存储在文件、PKCS #11 密钥库或 NSS 密钥库中。	第 208 页中的“如何使用 pktool 命令生成对称密钥”
提供可确保文件完整性的校验和。	检验接收者的文件副本是否与发送的文件完全相同。	第 212 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。	向消息接收者确认您是发送者。	第 213 页中的“如何计算文件的 MAC”
对文件加密，然后将加密文件解密。	通过对文件进行加密来保护文件内容。提供加密参数用于解密文件。	第 215 页中的“如何加密和解密文件”

▼ 如何使用 dd 命令生成对称密钥

加密文件以及生成文件的 MAC 时需要使用密钥。密钥应来源于随机数池。

要创建密钥，有三种选择：

- 如果您的站点上有随机数生成器，请使用此生成器。
- 如果要生成密钥并将其存储，请参见第 208 页中的“如何使用 pktool 命令生成对称密钥”。
- 否则，请使用以下过程。以下过程要求提供密钥大小（以位为单位）。与之相反，pktool 命令将根据您指定的算法确定正确的密钥大小。

1 确定算法所需的密钥长度。

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
sha256_hmac     8      512
sha384_hmac     8     1024
sha512_hmac     8     1024
```

b. 确定要传递到 dd 命令的密钥长度（以字节为单位）。

将最小最大密钥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。要生成随机密钥，请使用 /dev/urandom 文件。

of=keyfile 保存所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，请将密钥的位长度除以 8。

count=n 输入块的计数。*n* 的值应为 1。

3 将密钥存储在受保护的目录中。

密钥文件不应由除用户之外的任何人读取。

```
% chmod 400 keyfile
```

示例 12-1 创建用于 AES 算法的密钥

在下面的示例中，将创建一个用于 AES 算法的密钥。该密钥还将进行存储以供日后解密之用。AES 机制使用 128 位密钥。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 12-2 创建用于 DES 算法的密钥

在下面的示例中，将创建一个用于 DES 算法的密钥。该密钥还将进行存储以供日后解密之用。DES 机制使用 64 位密钥。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 12-3 创建用于 3DES 算法的密钥

在下面的示例中，将创建一个用于 3DES 算法的密钥。该密钥还将进行存储以供日后解密之用。3DES 机制使用 192 位密钥。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 12-4 创建用于 MD5 算法的密钥

在下面的示例中，将创建一个用于 MD5 算法的密钥。该密钥还将进行存储以供日后解密之用。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
```

▼ 如何使用 `pktool` 命令生成对称密钥

某些应用程序要求使用对称密钥对通信进行加密和解密。在此过程中，应创建一个对称密钥并存储它。

- 如果您的站点有随机数生成器，可使用此生成器为该密钥创建随机数。此过程不使用您站点的随机数生成器。
- 您可以使用以 `/dev/urandom` 设备作为输入的 `dd` 命令。`dd` 命令不会存储密钥。有关此过程，请参见第 206 页中的“如何使用 `dd` 命令生成对称密钥”。

1 可选如果计划使用密钥库，需要创建一个。

- 有关创建和初始化 PKCS #11 密钥库的信息，请参见第 238 页中的“如何使用 `pktool setpin` 命令生成口令短语”。
- 有关创建和初始化 NSS 数据库的信息，请参见示例 13-5。

2 生成一个随机数以用作对称密钥。

使用以下方法之一。

- 生成一个密钥并将其存储于文件中。

以文件存储的密钥的优点是，您可以从该文件提取密钥以用于应用程序的密钥文件中，如 `/etc/inet/secret/ipseckeys` 文件或 IPsec。

```
% pktool genkey keystore=file outkey=key-fn \
[keytype=generic|specific-symmetric-algorithm] [keylen=size-in-bits] \
[dir=directory] [print=n]
```

`keystore`

值 `file` 指定密钥存储位置的文件类型。

`outkey=key-fn`

`keystore=file` 时的文件名。

`keytype=specific-symmetric-algorithm`

无论对称密钥为何种长度，值均为 `generic`。对于特定算法，指定 `aes`、`arcfour`、`des` 或 `3des`。

`keylen=size-in-bits`

密钥的长度（以位为单位）。该数字必须可以被 8 整除。请勿对 `des` 或 `3des` 指定此值。

`dir=directory`

`key-fn` 所在的目录路径。缺省情况下，`directory` 是当前目录。

`print=n`

将密钥显示至终端窗口。缺省情况下，`print` 的值为 `n`。

- 生成一个密钥并将其存储于 PKCS #11 密钥库中。

PKCS #11 密钥库的优点是，您可以按密钥标签检索密钥。此方法适用于加密和解密文件所用的密钥。在使用此方法之前，必须先完成 [步骤 1](#)。

```
% pktool genkey label=key-label \
[keytype=generic|specific-symmetric-algorithm] [keylen=size-in-bits] \
[token=token] [sensitive=n] [extractable=y] [print=n]
```

`label=key-label`

密钥的用户指定标签。可以从密钥库中按密钥标签检索密钥。

`keytype=specific-symmetric-algorithm`

无论对称密钥为何种长度，值均为 `generic`。对于特定算法，指定 `aes`、`arcfour`、`des` 或 `3des`。

`keylen=size-in-bits`

密钥的长度（以位为单位）。该数字必须可以被 8 整除。请勿对 `des` 或 `3des` 指定此值。

`token=token`

令牌名称。缺省情况下，令牌是 `Sun Software PKCS#11 softtoken`。

`sensitive=n`

指定密钥的敏感度。当此值为 `y` 时，不可使用 `print=y` 参数显示密钥。缺省情况下，`sensitive` 的值为 `n`。

`extractable=y`

指定可从密钥库提取密钥。指定 `n` 可阻止提取密钥。

`print=n`

将密钥显示至终端窗口。缺省情况下，`print` 的值为 `n`。

- 生成一个密钥并将其存储于 NSS 密钥库中。

在使用此方法之前，必须先完成 [步骤 1](#)。

```
% pktool keystore=nss genkey label=key-label \
[keytype=[keytype=generic|specific-symmetric-algorithm] [keylen=size-in-bits] [token=token] \
[dir=directory-path] [prefix=database-prefix]
```

`keystore`

值 `nss` 指定密钥存储位置的 NSS 类型。

`label=key-label`

密钥的用户指定标签。可以从密钥库中按密钥标签检索密钥。

`keytype=symmetric-symmetric-algorithm`

无论对称密钥为何种长度，值均为 `generic`。对于特定算法，指定 `aes`、`arcfour`、`des` 或 `3des`。

`keylen=size-in-bits`

密钥的长度（以位为单位）。该数字必须可以被 8 整除。请勿对 `des` 或 `3des` 指定此值。

`token=token`

令牌名称。缺省情况下，该令牌为 NSS 内部令牌。

`dir=directory`

NSS 数据库所在的目录路径。缺省情况下，`directory` 是当前目录。

`prefix=directory`

NSS 数据库的前缀。缺省为无前缀。

`print=n`

将密钥显示至终端窗口。缺省情况下，`print` 的值为 `n`。

3 可选检验密钥是否存在。

根据密钥的存储位置，使用以下命令之一。

■ 检验 `key-fn` 文件中的密钥。

```
% pktool list keystore=file objtype=key infile=key-fn
Found n keys.
Key #1 - keytype:location (keylen)
```

■ 检验 PKCS #11 或 NSS 密钥库中的密钥。

```
$ pktool list objtype=key
Enter PIN for keystore:
Found n keys.
Key #1 - keytype:location (keylen)
```

示例 12-5 使用 `pktool` 命令创建对称密钥

在以下示例中，用户首先创建一个 PKCS #11 密钥库，随后为应用程序生成一个较大的对称密钥。最后，用户验证密钥是否位于密钥库中。

```
# pktool setpin
Create new passphrase:    easily-remembered-hard-to-detect-password
Re-enter new passphrase:  Retype password
Passphrase changed.
% pktool genkey label=specialappkey keytype=generic keylen=1024
Enter PIN for Sun Software PKCS#11 softtoken :    Type password
```

```
% pktool list objtype=key
Enter PIN for Sun Software PKCS#11 softtoken : Type password

Found 1 keys.
Key #1 - symmetric: specialappkey (1024 bits)
```

示例 12-6 使用 pktool 命令创建 DES 密钥

在下面的示例中，将创建一个用于 DES 算法的密钥。该密钥存储在本地文件中以供日后解密之用。该命令使用 **400** 权限保护文件。创建了密钥后，**print=y** 选项将在终端窗口显示生成的密钥。

DES 机制使用 64 位密钥。拥有密钥文件的用户将使用 **od** 命令检索密钥。

```
% pktool genkey keystore=file outkey=64bit.file1 keytype=des print=y
Key Value ="a3237b2c0a8ff9b3"
% od -x 64bit.file1
00000000 a323 7b2c 0a8f f9b3
```

示例 12-7 为 IPsec 安全关联创建对称密钥

在以下示例中，管理员手动为 IPsec SA 创建密钥材料并将其存储在文件中。然后，管理员将密钥复制到 `/etc/inet/secret/ipseckeys` 文件，并销毁原始文件。

- 首先，管理员创建并显示 IPsec 策略要求的密钥：

```
# pktool genkey keystore=file outkey=ipencrin1 keytype=generic keylen=192 print=y
Key Value ="294979e512cb8e79370dabecadc3fcbb849e78d2d6bd2049"
# pktool genkey keystore=file outkey=ipencrout1 keytype=generic keylen=192 print=y
Key Value ="9678f80e33406c86e3d1686e50406bd0434819c20d09d204"
# pktool genkey keystore=file outkey=ipspi1 keytype=generic keylen=32 print=y
Key Value ="acbeaa20"
# pktool genkey keystore=file outkey=ipspi2 keytype=generic keylen=32 print=y
Key Value ="19174215"
# pktool genkey keystore=file outkey=ipsha21 keytype=generic keylen=256 print=y
Key Value ="659c20f2d6c3f9570bcee93e96d95e2263aca4eeb3369f72c5c786af4177fe9e"
# pktool genkey keystore=file outkey=ipsha22 keytype=generic keylen=256 print=y
Key Value ="b041975a0e1fce0503665c3966684d731fa3dbb12fcf87b0a837b2da5d82c810"
```

- 然后，管理员创建以下 `/etc/inet/secret/ipseckeys` 文件：

```
## SPI values require a leading 0x.
## Backslashes indicate command continuation.
##
## for outbound packets on this system
add esp spi 0xacbeaa20 \
src 192.168.1.1 dst 192.168.2.1 \
encr_alg aes auth_alg sha256 \
encrkey 294979e512cb8e79370dabecadc3fcbb849e78d2d6bd2049 \
authkey 659c20f2d6c3f9570bcee93e96d95e2263aca4eeb3369f72c5c786af4177fe9e
##
## for inbound packets
add esp spi 0x19174215 \
src 192.168.2.1 dst 192.168.1.1 \
```

```
encr_alg aes auth_alg sha256 \
encrkey 9678f80e33406c86e3d1686e50406bd0434819c20d09d204 \
authkey b041975a0e1fce0503665c3966684d731fa3dbb12fcf87b0a837b2da5d82c810
```

- 在验证 ipseckey 文件的语法有效之后，管理员将销毁原始密钥文件。

```
# ipseckey -c /etc/inet/secret/ipseckey
# rm ipencrin1 ipencrout1 ipspi1 ipspi2 ipsha21 ipsha22
```

- 管理员通过使用 ssh 命令或其他安全机制，将 ipseckey 文件复制到通信系统。在通信系统中，将反转保护。ipseckey 文件中的第一个条目保护传入数据包，第二个条目保护传出数据包。通信系统中没有生成任何密钥。

▼ 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检查文件是否被篡改。摘要不会修改原始文件。

1 列出可用摘要算法。

```
% digest -l
md5
sha1
sha256
sha384
sha512
```

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 [步骤 1](#) 的输出中显示的算法。

input-file digest 命令的输入文件。

digest-listing digest 命令的输出文件。

示例 12-8 使用 MD5 机制计算摘要

在下面的示例中，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，摘要保存时不会包含随附信息：


```
% digest -a md5 email.attach >> $HOME/digest.emails.05.07
% cat ~/digest.emails.05.07
85c0a53d1a5cc71ea34d9ee7b1b28b01
```

示例 12-9 使用 SHA1 机制计算摘要

在下面的示例中，`digest` 命令使用 SHA1 机制提供目录列表。结果存放于文件中。

```
% 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 (message authentication code, MAC) 可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
sha256_hmac    8       512
sha384_hmac    8      1024
sha512_hmac    8      1024
```

2 生成长度适当的对称密钥。

您有两种选择。可以提供生成密钥时所基于的 [passphrase](#)（口令短语）。或者，可以直接提供密钥。

- 如果提供口令短语，则必须存储或记住该口令短语。如果您联机存储口令短语，则只有您才可以读取该口令短语文件。
- 如果提供密钥，则其大小必须是适用于相应机制的正确大小。有关此过程，请参见 [第 206 页](#) 中的“如何使用 `dd` 命令生成对称密钥”。您也可以使用 `pktool` 命令。有关该过程和一些示例，请参见 [第 208 页](#) 中的“如何使用 `pktool` 命令生成对称密钥”。

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 | -K key-label [-T token]] input-file
```

`-v` 显示以下格式的输出：

	<i>algorithm (input-file) = mac</i>
-a algorithm	用于计算 MAC 的算法。键入 <code>mac -l</code> 命令的输出中显示的算法。
-k keyfile	包含算法所指定长度的密钥的文件。
-K key-label	PKCS #11 密钥库中的密钥标签。
-T token	令牌名称。缺省情况下，令牌是 Sun Software PKCS#11 softtoken。只有在使用 <code>-K key-label</code> 选项时才可使用。
<i>input-file</i>	MAC 的输入文件。

示例 12-10 使用 DES_MAC 和口令短语计算 MAC

在下面的示例中，将使用 DES_MAC 机制和基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 passphrase:      <Type passphrase>
des_mac (email.attach) = dd27870a
% echo "des_mac (email.attach) = dd27870a" >> ~/desmac.daily.05.07
```

示例 12-11 使用 MD5_HMAC 和密钥文件计算 MAC

在下面的示例中，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 12-12 使用 SHA1_HMAC 和密钥文件计算 MAC

在下面的示例中，将使用 SHA1_HMAC 机制和密钥对目录清单进行验证。结果存放于文件中。

```
% 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
```

示例 12-13 使用 SHA1_HMAC 和密钥标签计算 MAC。

在下面的示例中，将使用 SHA1_HMAC 机制和密钥对目录清单进行验证。结果将放置在用户的 PKCS #11 密钥库中。用户最初通过使用 `pktool setpin` 命令创建了密钥库及其口令。

```
% mac -a sha1_hmac -K legaldocs0507 docs/*
Enter pin for Sun Software PKCS#11 softtoken:    Type password
```

要从密钥库中检索 MAC，用户应使用详细选项，并提供密钥标签和通过验证的目录名称。

```
% mac -v -a sha1_hmac -K legaldocs0507 docs/*
Enter pin for Sun Software PKCS#11 softtoken:    Type password
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，请参见示例后面的小节。

1 创建长度适当的对称密钥。

您有两种选择。可以提供生成密钥时所基于的 [passphrase（口令短语）](#)。或者，可以直接提供密钥。

- 如果提供口令短语，则必须存储或记住该口令短语。如果您联机存储口令短语，则只有您才可以读取该口令短语文件。
- 如果提供密钥，则其大小必须是适用于相应机制的正确大小。有关此过程，请参见 [第 206 页中的“如何使用 dd 命令生成对称密钥”](#)。您也可以使用 `pktool` 命令。有关该过程和一些示例，请参见 [第 208 页中的“如何使用 pktool 命令生成对称密钥”](#)。

2 对文件进行加密。

提供密钥并使用 `encrypt` 命令及对称密钥算法参数。

```
% encrypt -a algorithm [-v] \
[-k keyfile | -K key-label [-T token]] [-i input-file] [-o output-file]
```

- a *algorithm* 用于加密文件的算法。键入 `encrypt -l` 命令输出中显示的算法。
- 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）在 `encrypt -l` 命令的输出中列出。
- K *key-label* PKCS #11 密钥库中的密钥标签。

- T *token* 令牌名称。缺省情况下，令牌是 Sun Software PKCS#11 softtoken。只有在在使用 -K *key-label* 选项时才可使用。
- i *input-file* 要加密的输入文件。该命令不会更改此文件。
- o *output-file* 输出文件，是加密形式的输入文件。

示例 12-14 创建 AES 密钥以加密文件

在以下示例中，用户创建 AES 密钥并将其存储在现有 PKCS #11 密钥库中以用于加密和解密。用户可以验证密钥是否存在，也可以使用密钥，但是无法查看密钥本身。

```
% pktool genkey label=MyAESkeynumber1 keytype=aes keylen=256
Enter PIN for Sun Software PKCS#11 softtoken :     Type password
```

```
% pktool list objtype=key
Enter PIN for Sun Software PKCS#11 softtoken :<Type password>
Found 1 key
Key #1 - Sun Software PKCS#11 softtoken: MyAESkeynumber1 (256)
```

要使用密钥来加密文件，用户应通过密钥标签检索密钥。

```
% encrypt -a aes -K MyAESkeynumber1 -i encryptthisfile -o encryptedthisfile
```

要解密 encryptedthisfile 文件，用户应通过密钥标签检索密钥。

```
% decrypt -a aes -K MyAESkeynumber1 -i encryptedthisfile -o sameasencryptthisfile
```

示例 12-15 使用 AES 和口令短语进行加密和解密

在下面的示例中，将使用 AES 算法来加密文件。密钥基于口令短语生成。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 passphrase:     <Type passphrase>
Re-enter passphrase:     Type passphrase again
```

输入文件 ticket.to.ride 仍然以其原始形式存在。

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 passphrase:     <Type passphrase>
```

示例 12-16 使用 AES 和密钥文件进行加密和解密

在下面的示例中，将使用 AES 算法来加密文件。AES 机制使用 128 位（即 16 字节）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 仍然以其原始形式存在。

要解密输出文件，用户应使用加密文件时所用的密钥和加密机制。

```
% decrypt -a aes -k ~/keyf/05.07.aes16 \  
-i ~/enc/e.ticket.to.ride -o ~/d.ticket.to.ride
```

示例 12-17 使用 ARCFOUR 和密钥文件进行加密和解密

在下面的示例中，将使用 ARCFOUR 算法来加密文件。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，用户应使用加密文件时所用的密钥和加密机制。

```
% decrypt -a arcfour -i ~/enc/e.personal.txt \  
-k ~/keyf/05.07.rc4.8 -o ~/personal.txt
```

示例 12-18 使用 3DES 和密钥文件进行加密和解密

在下面的示例中，将使用 3DES 算法来加密文件。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，用户应使用加密文件时所用的密钥和加密机制。

```
% decrypt -a 3des -k ~/keyf/05.07.des24 \  
-i ~/enc/e.personal2.txt -o ~/personal2.txt
```

故障排除 以下消息表明，所使用的算法不接受您提供给 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

如果传递的密钥不满足算法的要求，您必须提供正确的密钥。

- 一种选择是使用口令短语。然后，框架提供满足要求的密钥。
- 第二种选择是传递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）

本节介绍如何管理加密框架中的软件提供者和硬件提供者。如果需要，可以禁用软件提供者和硬件提供者。例如，可以禁用某个软件提供者的算法实现。然后，可以强制系统使用其他软件提供者的算法。

管理加密框架（任务列表）

以下任务列表列出了管理加密框架中的软件和硬件提供者的过程。

任务	说明	参考
列出加密框架中的提供者。	列出可在加密框架中使用的算法、库和硬件设备。	第 218 页中的“如何列出可用提供者”
添加软件提供者。	将 PKCS #11 库或内核模块添加到加密框架中。必须对提供者进行签名。	第 222 页中的“如何添加软件提供者”
禁止使用用户级机制。	禁用软件机制。可以再次启用该机制。	第 223 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。	临时禁用某个机制。通常用于测试。	第 225 页中的“如何禁止使用内核软件提供者”
卸载提供者。	禁用内核软件提供者。	示例 12-27
列出可用硬件提供者。	显示连接的硬件、该硬件提供的机制以及启用的机制。	第 227 页中的“如何列出硬件提供者”
禁用硬件提供者的机制。	确保未使用硬件加速器的所选机制。	第 228 页中的“如何禁用硬件提供者机制和功能”
重新启动或刷新加密服务。	确保加密服务可用。	第 230 页中的“如何刷新或重新启动所有加密服务”

▼ 如何列出可用提供者

加密框架可为多种类型的使用者提供算法：

- 用户级提供者与 libpkcs11 库相链接的应用程序提供 PKCS #11 加密接口
- 内核软件提供者与 IPsec、Kerberos 及其他 Oracle Solaris 内核组件提供算法
- 内核硬件提供者通过 pkcs11_kernel 库提供内核使用者和应用程序可用的算法

1 以简要格式列出提供者。

注 – 不同的 Oracle Solaris 发行版中提供者列表的内容和格式会有所不同。在您的系统上运行 `cryptoadm list` 命令可查看系统支持的提供者。

一般用户只能使用用户级机制。

```
% cryptoadm list
User-level providers:
Provider: /usr/lib/security/$ISA/pkcs11_kernel.so
Provider: /usr/lib/security/$ISA/pkcs11_softtoken.so
Provider: /usr/lib/security/$ISA/pkcs11_tpm.so

Kernel software providers:
  des
  aes
  arcfour
  blowfish
  ecc
  sha1
  sha2
  md4
  md5
  rsa
  swrand

Kernel hardware providers:
  ncp/0
```

2 列出加密框架中的提供者及其机制。

以下输出中列出了所有机制。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 12-20。

为了便于显示，输出将被截断。

```
% cryptoadm list -m
User-level providers:
=====

Provider: /usr/lib/security/$ISA/pkcs11_kernel.so
/usr/lib/security/$ISA/pkcs11_kernel.so: no slots presented.

Provider: /usr/lib/security/$ISA/pkcs11_softtoken.so
Mechanisms:
CKM_DES_CBC
CKM_DES_CBC_PAD
CKM_DES_ECB
CKM_DES_KEY_GEN
CKM_DES_MAC_GENERAL
...
CKM_ECDSA_SHA1
CKM_ECDH1_DERIVE

Provider: /usr/lib/security/$ISA/pkcs11_tpm.so
/usr/lib/security/$ISA/pkcs11_tpm.so: no slots presented.
```

```

Kernel software providers:
=====
des: CKM_DES_ECB,CKM_DES_CBC,CKM_DES3_ECB,CKM_DES3_CBC
aes: CKM_AES_ECB,CKM_AES_CBC,CKM_AES_CTR,CKM_AES_CCM,CKM_AES_GCM,CKM_AES_GMAC
arcfour: CKM_RC4
blowfish: CKM_BLOWFISH_ECB,CKM_BLOWFISH_CBC
ecc: CKM_EC_KEY_PAIR_GEN,CKM_ECDH1_DERIVE,CKM_ECDSA,CKM_ECDSA_SHA1
sha1: CKM_SHA_1,CKM_SHA_1_HMAC,CKM_SHA_1_HMAC_GENERAL
sha2: CKM_SHA256,CKM_SHA256_HMAC,CKM_SHA256_HMAC_GENERAL,CKM_SHA384,CKM_SHA384_HMAC,
CKM_SHA384_HMAC_GENERAL,CKM_SHA512,CKM_SHA512_HMAC,CKM_SHA512_HMAC_GENERAL
md4: CKM_MD4
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,
CKM_SHA256_RSA_PKCS,CKM_SHA384_RSA_PKCS,CKM_SHA512_RSA_PKCS
swrand: No mechanisms presented.

Kernel hardware providers:
=====
ncp/0: CKM_DSA,CKM_RSA_X_509,CKM_RSA_PKCS,CKM_RSA_PKCS_KEY_PAIR_GEN,
CKM_DH_PKCS_KEY_PAIR_GEN,CKM_DH_PKCS_DERIVE,CKM_EC_KEY_PAIR_GEN,
CKM_ECDH1_DERIVE,CKM_ECDSA

```

示例 12-19 查找现有加密机制

在下面的示例中，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
Mechanisms:
CKM_DES_CBC
CKM_DES_CBC_PAD
CKM_DES_ECB
CKM_DES_KEY_GEN
CKM_DES_MAC_GENERAL
CKM_DES_MAC
...
CKM_ECDSA
CKM_ECDSA_SHA1
CKM_ECDH1_DERIVE

```

示例 12-20 查找可用加密机制

策略确定可使用的机制。管理员负责设置策略。管理员可以选择禁用特定提供者的机制。-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.
/usr/lib/security/\$ISA/pkcs11_softtoken.so: all mechanisms are enabled,
except CKM_MD5. random is enabled.
/usr/lib/security/\$ISA/pkcs11_tpm.so: all mechanisms are enabled.

Kernel software providers:
=====

```



```

des: all mechanisms are enabled.
aes: all mechanisms are enabled.
arcfour: all mechanisms are enabled.
blowfish: all mechanisms are enabled.
ecc: all mechanisms are enabled.
sha1: all mechanisms are enabled.
sha2: all mechanisms are enabled.
md4: all mechanisms are enabled.
md5: all mechanisms are enabled.
rsa: all mechanisms are enabled.
swrand: random is enabled.

Kernel hardware providers:
=====
ncp/0: all mechanisms are enabled. random is enabled.

```

示例 12-21 确定哪些加密机制执行哪些功能

机制执行特定加密功能，如签名或密钥生成。-v -m 选项显示每个机制及其功能。

在本实例中，管理员要确定 CKM_ECDSA* 机制可用于哪些功能。

```

% cryptoadm list -vm
User-level providers:
=====

Provider: /usr/lib/security/$ISA/pkcs11_kernel.so
/usr/lib/security/$ISA/pkcs11_kernel.so: no slots presented.

Provider: /usr/lib/security/$ISA/pkcs11_softtoken.so
...
CKM_ECDSA      112 571 . . . . X . X . . . . .
CKM_ECDSA_SHA1 112 571 . . . . X . X . . . . .
...

```

该列表指示这些用户级机制可从 /usr/lib/security/\$ISA/pkcs11_softtoken.so 库获得。

条目中的每一项代表机制的一条相关信息。对于这些 ECC 机制，该列表指示以下内容：

- 最小长度—112 字节
- 最大长度—571 字节
- 硬件—硬件不可用。
- 加密—不可用于加密数据。
- 解密—不可用于解密数据。
- 摘要—不可用于创建消息摘要。
- 签名—用于对数据签名。
- 签名+恢复—不可用于对能从签名中恢复的数据签名。
- 验证—用于验证签名数据。
- 验证+恢复—不可用于验证能从签名中恢复的数据。
- 密钥生成—不可用于生成私钥。

- 密钥对生成—不可用于生成密钥对。
- 包装—不可用于包装。即加密，现有密钥。
- 解包—不可用于将已包装的密钥解包。
- 派生—不可用于从基本密钥派生新密钥。

▼ 如何添加软件提供者

开始之前 您必须分配有 Crypto Management（加密管理）权限配置文件。

1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 列出系统可用的软件提供者。

```
% cryptoadm list
User-level providers:
Provider: /usr/lib/security/$ISA/pkcs11_kernel.so
Provider: /usr/lib/security/$ISA/pkcs11_softtoken.so
/usr/lib/security/$ISA/pkcs11_tpm.so: all mechanisms are enabled.
```

Kernel software providers:

```
des
aes
arcfour
blowfish
sha1
sha2
md4
md5
rsa
swrand
```

Kernel hardware providers:

```
ncp/0
```

3 从系统信息库添加提供者。

Oracle 已向现有提供者软件颁发证书。

4 刷新提供者。

如果添加了软件提供者或者添加了硬件并为该硬件指定了策略，则需要刷新提供者。

```
# svcadm refresh svc:/system/cryptosvc
```

5 在列表中找到新增的提供者。

在本例中，安装了一个新的内核软件提供者。

```
# cryptoadm list
...
Kernel software providers:
des
aes
```

```

    arcfour
    blowfish
    ecc
    sha1
    sha2
    md4
    md5
    rsa
    swrand
    sha3      <-- added provider
...

```

示例 12-22 添加用户级软件提供者

在下面的示例中，将安装签名的 PKCS #11 库。

```

# pkgadd -d /cdrom/cdrom0/SolarisNew
  Answer the prompts
# svcadm refresh system/cryptosvc
# cryptoadm list
user-level providers:
=====
    /usr/lib/security/$ISA/pkcs11_kernel.so
    /usr/lib/security/$ISA/pkcs11_softtoken.so
    /usr/lib/security/$ISA/pkcs11_tpm.so
    /opt/lib/$ISA/libpkcs11.so.1      <-- added provider

```

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/lib/$ISA/libpkcs11.so.1

```

▼ 如何禁止使用用户级机制

如果不应使用库提供者的某些加密机制，可以删除这些所选的机制。此过程以 pkcs11_softtoken 库中的 DES 机制为例。

开始之前 您必须分配有 Crypto Management（加密管理）权限配置文件。

- 1 成为具有所需安全属性的管理员。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。
- 2 列出特定用户级软件提供者提供的机制。

```

% 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 列出可使用的机制。

```
$ cryptoadm list -p
user-level providers:
=====
...
/usr/lib/security/$ISA/pkcs11_softtoken.so: all mechanisms are enabled.
random is enabled.
...
```

4 禁用不应使用的机制。

```
$ 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 12-23 启用一个用户级软件提供者机制

在下面的示例中，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 12-24 启用所有用户级软件提供者机制

在下面的示例中，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 12-25 永久禁用用户级软件提供者

在下面的示例中，将删除 libpkcs11.so.1 库。

```
$ cryptoadm uninstall provider=/opt/lib/\$ISA/libpkcs11.so.1
$ cryptoadm list
user-level providers:
```

```

/usr/lib/security/$ISA/pkcs11_kernel.so
/usr/lib/security/$ISA/pkcs11_softtoken.so
/usr/lib/security/$ISA/pkcs11_tpm.so

kernel software providers:
...
```

▼ 如何禁止使用内核软件提供者

如果加密框架提供了某个提供者（如 AES）的多种模式，则可以禁用速度较慢的机制或被破坏的机制。此过程以 AES 算法为例。

开始之前 您必须分配有 Crypto Management（加密管理）权限配置文件。

- 1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

- 2 列出特定内核软件提供者提供的机制。

```

$ cryptoadm list -m provider=aes
aes: CKM_AES_ECB,CKM_AES_CBC,CKM_AES_CTR,CKM_AES_CCM,CKM_AES_GCM,CKM_AES_GMAC
```

- 3 列出可使用的机制。

```

$ cryptoadm list -p provider=aes
aes: all mechanisms are enabled.
```

- 4 禁用不应使用的机制。

```

$ cryptoadm disable provider=aes mechanism=CKM_AES_ECB
```

- 5 列出可使用的机制。

```

$ cryptoadm list -p provider=aes
aes: all mechanisms are enabled, except CKM_AES_ECB.
```

示例 12-26 启用内核软件提供者机制

在下面的示例中，将使禁用的 AES 机制再次可用。

```

cryptoadm list -m provider=aes
aes: CKM_AES_ECB,CKM_AES_CBC,CKM_AES_CTR,CKM_AES_CCM,CKM_AES_GCM,CKM_AES_GMAC
$ 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 12-27 临时禁用内核软件提供者

在下面的示例中，将临时禁用 AES 提供者。unload 子命令可用于禁止在卸载某个提供者时自动装入该提供者。例如，安装影响提供者的修补程序时，将使用 unload 子命令。

```
$ cryptoadm unload provider=aes
```

```
$ cryptoadm list
```

```
...
Kernel software providers:
  des
  aes (inactive)
  arcfour
  blowfish
  ecc
  sha1
  sha2
  md4
  md5
  rsa
  swrand
```

AES 提供者只有在加密框架刷新之后才会可用。

```
$ svcadm refresh system/cryptosvc
```

```
$ cryptoadm list
```

```
...
Kernel software providers:
  des
  aes
  arcfour
  blowfish
  ecc
  sha1
  sha2
  md4
  md5
  rsa
  swrand
```

如果内核使用者正在使用内核软件提供者，则不会卸载该软件。此时将显示错误消息，不过该提供者仍然可供使用。

示例 12-28 永久禁用软件提供者

在下面的示例中，将禁用 AES 提供者。禁用后，该 AES 提供者将不会出现在内核软件提供者的策略列表中。

```
$ cryptoadm uninstall provider=aes
```

```
$ cryptoadm list
...
Kernel software providers:
  des
  arcfour
  blowfish
  ecc
  sha1
  sha2
  md4
  md5
  rsa
  swrand
```

如果内核使用者正在使用内核软件提供者，将显示一条错误消息，不过该提供者仍然可供使用。

示例 12-29 重新安装已删除的内核软件提供者

在下面的示例中，将重新安装 AES 内核软件提供者。

```
$ cryptoadm install provider=aes \
mechanism=CKM_AES_ECB,CKM_AES_CBC,CKM_AES_CTR,CKM_AES_CCM,CKM_AES_GCM,CKM_AES_GMAC

$ cryptoadm list
...
Kernel software providers:
  des
  aes
  arcfour
  blowfish
  ecc
  sha1
  sha2
  md4
  md5
  rsa
  swrand
```

▼ 如何列出硬件提供者

硬件提供者将自动定位和装入。有关更多信息，请参见 [driver.conf\(4\)](#) 手册页。

开始之前 如果您的硬件预计会在加密框架中使用，该硬件将通过 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 Oracle 颁发的证书对驱动程序的目标文件进行签名。

例如，Sun Crypto Accelerator 6000 板 (mca)、UltraSPARC T1 和 T2 处理器上加密加速器的 ncp 驱动程序 (ncp) 以及 UltraSPARC T2 处理器的 n2cp 驱动程序 (n2cp) 将硬件机制插入到框架中。

有关对提供者进行签名的信息，请参见第 203 页中的“第三方软件的二进制文件签名”。

1 列出系统上可用的硬件提供者。

```
% cryptoadm list
...
kernel hardware providers:
  ncp/0
```

2 列出芯片或板提供的机制。

```
% cryptoadm list -m provider=ncp/0
ncp/0:
CKM_DSA
CKM_RSA_X_509
...
CKM_ECDH1_DERIVE
CKM_ECDSA
```

3 列出芯片或板上可用的机制。

```
% cryptoadm list -p provider=ncp/0
ncp/0: all mechanisms are enabled.
```

▼ 如何禁用硬件提供者机制和功能

可以有选择地禁用硬件提供者的机制和随机数功能。有关重新启用它们的信息，请参见示例 12-30。此示例中的硬件 Sun Crypto Accelerator 1000 板提供了一个随机数生成器。

开始之前 您必须分配有 Crypto Management（加密管理）权限配置文件。

1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 选择要禁用的机制或功能。

列出硬件提供者。

```
# cryptoadm list
...
Kernel hardware providers:
  dca/0
```

■ 禁用选定的机制。

```
# cryptoadm list -m provider=dca/0
dca/0: CKM_RSA_PKCS, CKM_RSA_X_509, CKM_DSA, CKM_DES_CBC, CKM_DES3_CBC
random is enabled.
# cryptoadm disable provider=dca/0 mechanism=CKM_DES_CBC,CKM_DES3_CBC
# cryptoadm list -p provider=dca/0
dca/0: all mechanisms are enabled except CKM_DES_CBC,CKM_DES3_CBC.
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.
```

- 禁用该硬件上所有功能和机制。

```
# 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 12-30 启用硬件提供者的机制和功能

在下面的示例中，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.
```

在下面的示例中，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.
```

在下面的示例中，将仅启用机制。随机数生成器将继续处于禁用状态。

```
# 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.
```

在下面的示例中，将启用板上的所有功能和机制。

```
# 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，将启用加密框架。无论 `kcf` 守护进程因何种原因失败，都可使用服务管理工具 (Service Management Facility, SMF) 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。有关重新启动加密服务对区域的影响，请参见第 204 页中的“加密服务和区域”。

开始之前 您必须分配有 Crypto Management（加密管理）权限配置文件。

- 1 成为具有所需安全属性的管理员。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。

- 2 检查加密服务的状态。

```
% svcs cryptosvc
STATE      STIME    FMRI
offline    Dec_09   svc:/system/cryptosvc:default
```

- 3 启用加密服务。

```
# svcadm enable svc:/system/cryptosvc
```

示例 12-31 刷新加密服务

在下面的示例中，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
```

密钥管理框架

Oracle Solaris 的密钥管理框架 (Key Management Framework, KMF) 功能提供了用于管理公钥对象的工具及编程接口。公钥对象包括 X.509 证书和公钥/私钥对。存储这些对象所用的格式可能有所不同。KMF 还提供了一种工具，用于管理定义应用程序如何使用 X.509 证书的策略。KMF 支持第三方插件。

- 第 231 页中的“管理公钥技术”
- 第 232 页中的“密钥管理框架实用程序”
- 第 233 页中的“使用密钥管理框架（任务）”

管理公钥技术

密钥管理框架 (Key Management Framework, KMF) 提供了一种统一管理公钥技术 (public key technology, PKI) 的方法。Oracle Solaris 中有多种不同的应用程序采用了 PKI 技术。每种应用程序都提供自己的编程接口、密钥存储机制和管理实用程序。如果某个应用程序提供了策略执行机制，则此机制将仅适用于该应用程序。借助 KMF，各应用程序可以使用一套统一的管理工具，单一一组编程接口，以及同一个策略执行机制。这些功能可以管理采用这些接口的所有应用程序的 PKI 需求。

KMF 通过以下接口统一管理公钥技术：

- **pktool 命令**—此命令管理各种密钥库中的 PKI 对象（如证书）。
- **kmfcfg 命令**—此命令管理 PKI 策略数据库和第三方插件。

PKI 策略决策包括一些操作，如某项操作的验证方法。此外，PKI 策略还可以限制证书的范围。例如，PKI 策略可以声明某个证书仅可用于特定的目的。这种策略可防止将证书用于其他请求。

- **KMF 库**—此库包含将底层密钥库机制抽象化的编程接口。

应用程序不必选择某种特定的密钥库机制，但可以从一种机制迁移到另一种机制。支持的密钥库包括 PKCS #11、NSS 和 OpenSSL。该库具有可插接式框架，因此可添加新的密钥库机制。由于这一原因，使用新机制的应用程序将只需要微小的修改就能使用新密钥库。

注 - 要确定正在运行的 OpenSSL 的版本，请键入 `openssl version`。输出以下类似内容：

```
OpenSSL 1.0.0d 8 Feb 2011
```

密钥管理框架实用程序

KMF 提供了管理密钥存储的方法，并且提供了使用这些密钥的整体策略。KMF 管理以下三种公钥技术的策略、密钥和证书：

- 来自 PKCS #11 提供者（即来自加密框架）的令牌
- NSS，即网络安全服务 (Network Security Services)
- OpenSSL，一种基于文件的密钥库

`kmfcfg` 工具可以创建、修改或删除 KMF 策略项。该工具还可以管理框架的插件。KMF 通过 `pktool` 命令管理密钥库。有关更多信息，请参见 [kmfcfg\(1\)](#) 和 [pktool\(1\)](#) 手册页以及以下各节。

KMF 策略管理

KMF 策略存储在数据库中。使用 KMF 编程接口的所有应用程序在内部访问此策略数据库。该数据库可以约束由 KMF 库管理的密钥和证书的使用。当某个应用程序试图验证证书时，该应用程序将检查策略数据库。使用 `kmfcfg` 命令可修改策略数据库。

KMF 插件管理

`kmfcfg` 命令提供了以下插件子命令：

- `list plugin`—列出由 KMF 管理的插件。
- `install plugin`—按模块的路径名安装插件并为插件创建密钥库。要从 KMF 删除插件，请删除密钥库。
- `uninstall plugin`—通过删除插件的密钥库从 KMF 删除插件。
- `modify plugin`—使插件通过在插件的代码中定义的选项（如 `debug`）运行。

有关更多信息，请参见 [kmfcfg\(1\)](#) 手册页。有关过程，请参见第 243 页中的“如何在 KMF 中管理第三方插件”。

KMF 密钥库管理

KMF 可管理以下三种公钥技术的密钥库：PKCS #11 令牌、NSS 和 OpenSSL。对于所有这些技术，pktool 命令可用于执行以下操作：

- 生成自签名证书。
- 生成证书请求。
- 生成对称密钥。
- 生成公钥/私钥对。
- 生成要发送到外部证书颁发机构 (certificate authority, CA) 进行签署的 PKCS #10 证书签名请求 (certificate signing request, CSR)。
- 签署 PKCS #10 CSR。
- 将对象导入密钥库。
- 列出密钥库中的对象。
- 删除密钥库中的对象。
- 下载 CRL。

对于 PKCS #11 和 NSS 技术，pktool 命令还可用于通过生成口令短语设置 PIN：

- 为密钥库生成口令短语。
- 为密钥库中的对象生成口令短语。

有关使用 pktool 实用程序的示例，请参见 [pktool\(1\)](#) 手册页以及第 233 页中的“使用密钥管理框架（任务列表）”。

使用密钥管理框架（任务）

本节介绍如何使用 pktool 命令管理公钥对象，例如口令、口令短语、文件、密钥库、证书和 CRL。

使用密钥管理框架（任务列表）

通过密钥管理框架 (Key Management Framework, KMF) 可以集中管理公钥技术。

任务	说明	参考
创建证书。	创建供 PKCS #11、NSS 或 SSL 使用的证书。	第 234 页中的“如何使用 pktool gencert 命令创建证书”
导出证书。	创建包含证书及其支持密钥的文件。可以使用口令保护该文件。	第 237 页中的“如何以 PKCS #12 格式导出证书和私钥”

任务	说明	参考
导入证书。	从其他系统导入证书。	第 235 页中的“如何将证书导入密钥库”
	从其他系统导入 PKCS #12 格式的证书。	示例 13-2
生成口令短语。	生成访问 PKCS #11 密钥库或 NSS 密钥库所用的口令短语。	第 238 页中的“如何使用 <code>pktool setpin</code> 命令生成口令短语”
生成对称密钥。	生成用于加密文件、创建文件的 MAC 和应用程序的对称密钥。	第 208 页中的“如何使用 <code>pktool</code> 命令生成对称密钥”
生成密钥对。	生成用于应用程序的公钥/私钥对。	第 238 页中的“如何使用 <code>pktool genkeypair</code> 命令生成密钥对”
生成 PKCS #10 CSR。	生成要外部证书颁发机构 (CA) 签署的 PKCS #10 证书签名请求 (certificate signing request, CSR)。	<code>pktool(1)</code> 手册页
签署 PKCS #10 CSR。	签署 PKCS #10 CSR。	第 242 页中的“如何使用 <code>pktool signcsr</code> 命令签署证书请求”
将插件添加到 KMF 中。	安装、修改和列出插件。此外，还从 KMF 中删除插件。	第 243 页中的“如何在 KMF 中管理第三方插件”

▼ 如何使用 `pktool gencert` 命令创建证书

此过程可创建自签名证书，并将该证书存储在 PKCS #11 密钥库中。在此操作过程中，还将创建一个 RSA 公钥/私钥对。私钥与该证书一起存储在密钥库中。

1 生成自签名证书。

```
% pktool gencert [keystore=keystore] label=label-name \  
subject=subject-DN serial=hex-serial-number
```

`keystore=keystore` 按公钥对象类型指定密钥库。该值可以是 `nss`、`pkcs11` 或 `ssl`。此关键字是可选的。

`label=label-name` 指定颁发者为证书提供的唯一名称。

`subject=subject-DN` 指定证书的标识名。

`serial=hex-serial-number` 指定十六进制格式序列号。证书的颁发者可以选择该序列号，例如 `0x0102030405`。

2 验证密钥库的内容。

```
% pktool list  
Found number certificates.  
1. (X.509 certificate)  
  Label: label-name  
  ID: Fingerprint that binds certificate to private key  
  Subject: subject-DN
```

```

        Issuer: distinguished-name
        Serial: hex-serial-number
n. ...

```

此命令可列出密钥库中的所有证书。在下面的示例中，密钥库只包含一个证书。

示例 13-1 使用 pktool 创建自签名证书

在下面的示例中，My Company 的用户创建了一个自签名证书，并将该证书存储在 PKCS #11 对象的密钥库中。该密钥库最初是空的。如果尚未初始化此密钥库，则软令牌的 PIN 为 changeme。

```

% pktool gencert keystore=pkcs11 label="My Cert" \
subject="C=US, O=My Company, OU=Security Engineering Group, CN=MyCA" \
serial=0x000000001
Enter pin for Sun Software PKCS#11 softtoken:    Type PIN for token

% pktool list
Found 1 certificates.
1. (X.509 certificate)
   Label: My Cert
   ID: 12:82:17:5f:80:78:eb:44:8b:98:e3:3c:11:c0:32:5e:b6:4c:ea:eb
   Subject: C=US, O=My Company, OU=Security Engineering Group, CN=MyCA
   Issuer: C=US, O=My Company, OU=Security Engineering Group, CN=MyCA
   Serial: 0x01

```

▼ 如何将证书导入密钥库

此过程介绍如何将包含以 PEM 或原始 DER 编码的 PKI 信息的文件导入密钥库。有关导出过程的信息，请参见[示例 13-4](#)。

1 导入证书。

```
% pktool import keystore=keystore infile=infile-name label=label-name
```

2 如果要导入私有 PKI 对象，需要在系统提示时提供口令。

a. 在提示符下提供文件的口令。

如果要导入的是私有 PKI 信息（例如 PKCS #12 格式的导出文件），则需要为该文件提供口令。您要导入的文件的创建者会向您提供 PKCS #12 口令。

```
Enter password to use for accessing the PKCS12 file:    Type PKCS#12 password
```

b. 在提示符下键入密钥库的口令。

```
Enter pin for Sun Software PKCS#11 softtoken:    Type PIN for token
```

3 检验密钥库的内容。

```
% pktool list
Found number certificates.
1. (X.509 certificate)
   Label: label-name
   ID: Fingerprint that binds certificate to private key
   Subject: subject-DN
   Issuer: distinguished-name
   Serial: hex-serial-number
2. ...
```

示例 13-2 将 PKCS #12 文件导入密钥库

在下面的示例中，用户从第三方导入了一个 PKCS #12 文件。pktool import 命令可从 gracedata.p12 文件提取私钥和证书，并将它们存储在用户的首选密钥库中。

```
% pktool import keystore=pkcs11 infile=gracedata.p12 label=GraceCert
Enter password to use for accessing the PKCS12 file: Type PKCS #12 password
Enter pin for Sun Software PKCS#11 softtoken: Type PIN for token
Found 1 certificate(s) and 1 key(s) in gracedata.p12
% pktool list
Found 1 certificates.
1. (X.509 certificate)
   Label: GraceCert
   ID: 12:82:17:5f:80:78:eb:44:8b:98:e3:3c:11:c0:32:5e:b6:4c:ea:eb
   Subject: C=US, O=My Company, OU=Security Engineering Group, CN=MyCA
   Issuer: C=US, O=My Company, OU=Security Engineering Group, CN=MyCA
   Serial: 0x01
```

示例 13-3 将 X.509 证书导入密钥库

在下面的示例中，用户将 PEM 格式的 X.509 证书导入用户的首选密钥库。此公共证书不受口令保护。用户的公共密钥库也不受口令保护。

```
% pktool import keystore=pkcs11 infile=somecert.pem label="TheirCompany Root Cert"
% pktool list
Found 1 certificates.
1. (X.509 certificate)
   Label: TheirCompany Root Cert
   ID: 21:ae:83:98:24:d1:1f:cb:65:5b:48:75:7d:02:47:cf:98:1f:ec:a0
   Subject: C=US, O=TheirCompany, OU=Security, CN=TheirCompany Root CA
   Issuer: C=US, O=TheirCompany, OU=Security, CN=TheirCompany Root CA
   Serial: 0x01
```


▼ 如何以 PKCS #12 格式导出证书和私钥

可以创建一个 PKCS #12 格式的文件，用于将私钥及其关联的 X.509 证书导出到其他系统。对该文件的访问受口令保护。

1 找到要导出的证书。

```
% pktool list
Found number certificates.
1. (X.509 certificate)
   Label: label-name
   ID: Fingerprint that binds certificate to private key
   Subject: subject-DN
   Issuer: distinguished-name
   Serial: hex-serial-number
2. ...
```

2 导出密钥和证书。

使用 `pktool list` 命令生成的密钥库和标签。为导出文件提供文件名。如果文件名包含空格，需要用双引号将其括起来。

```
% pktool export keystore=keystore outfile=outfile-name label=label-name
```

3 使用口令保护导出文件。

在提示符下键入密钥库的当前口令。此时，即会为导出文件创建口令。接收者在导入该文件时必须提供此口令。

```
Enter pin for Sun Software PKCS#11 softtoken: Type PIN for token
Enter password to use for accessing the PKCS12 file: Create PKCS #12 password
```

提示 – 将口令与导出文件分开发送。建议的最佳做法是不通过发送数据来提供口令，例如通过打电话。

示例 13–4 以 PKCS #12 格式导出证书和私钥

在下面的示例中，用户将私钥及其关联的 X.509 证书导出到一个标准的 PKCS #12 文件中。可将此文件导入到其他密钥库中。PKCS #11 口令可保护源密钥库。PKCS #12 口令用于保护 PKCS #12 文件中的私有数据。导入该文件时需要提供此口令。

```
% pktool list
Found 1 certificates.
1. (X.509 certificate)
   Label: My Cert
   ID: 12:82:17:5f:80:78:eb:44:8b:98:e3:3c:11:c0:32:5e:b6:4c:ea:eb
   Subject: C=US, O=My Company, OU=Security Engineering Group, CN=MyCA
   Issuer: C=US, O=My Company, OU=Security Engineering Group, CN=MyCA
   Serial: 0x01
```

```
% pktool export keystore=pkcs11 outfile=mydata.p12 label="My Cert"
Enter pin for Sun Software PKCS#11 softtoken:      Type PIN for token
Enter password to use for accessing the PKCS12 file:  Create PKCS #12 password
```

然后，用户打电话给接收者，提供 PKCS #12 口令。

▼ 如何使用 **pktool setpin** 命令生成口令短语

可以为密钥库中的对象以及密钥库本身生成口令短语。访问该对象或密钥库时需要提供此口令短语。有关为密钥库中的对象生成口令短语的示例，请参见[示例 13-4](#)。

1 生成访问密钥库所用的口令短语。

```
% pktool setpin keystore=nss|pkcs11 dir=directory
```

2 应答提示。

如果尚未为密钥库设置口令，请按回车键创建口令。

```
Enter current token passphrase:      Press the Return key
Create new passphrase:               Type the passphrase that you want to use
Re-enter new passphrase:             Retype the passphrase
Passphrase changed.
```

密钥库现在受**口令短语**保护了。如果丢失了口令短语，您将无法访问该密钥库中的对象。

示例 13-5 使用口令短语保护密钥库

下面的示例说明了如何为 NSS 数据库设置口令短语。由于尚未创建口令短语，用户需要在第一个提示处按回车键。

```
% pktool setpin keystore=nss dir=/var/nss
Enter current token passphrase:      Press the Return key
Create new passphrase:               has8n0NdaH
Re-enter new passphrase:             has8n0NdaH
Passphrase changed.
```

▼ 如何使用 **pktool genkeypair** 命令生成密钥对

某些应用程序需要公钥/私钥对。在此过程中，将创建这些密钥对并存储它们。

1 可选如果计划使用密钥库，请创建密钥库。

- 有关创建和初始化 PKCS #11 密钥库的信息，请参见[第 238 页中的“如何使用 **pktool setpin** 命令生成口令短语”](#)。

- 要创建和初始化 NSS 密钥库，请参见[示例 13-5](#)。

2 创建密钥对。

使用以下方法之一。

- 创建密钥对，然后将该密钥对存储在文件中。

基于文件的密钥是为从磁盘上的文件中直接读取密钥的应用程序而创建的。通常，直接使用 OpenSSL 加密库的应用程序要求您将应用程序的密钥和证书存储在文件中。

注 – file 密钥库不支持椭圆曲线 (elliptic curve, ec) 密钥和证书。

```
% pktool genkeypair keystore=file outkey=key-filename \
[format=der|pem] [keytype=rsa|dsa] [keylen=key-size]
```

keystore=file

值 file 指定密钥存储位置的文件类型。

outkey=key-filename

指定存储密钥对的文件的名称。

format=der|pem

指定密钥对的编码格式。der 输出是二进制的，pem 输出为 ASCII。

keytype=rsa|dsa

指定可存储在 file 密钥库中的密钥对的类型。有关定义，请参见[DSA](#)和[RSA](#)。

keylen=key-size

指定密钥的长度（以位为单位）。该数字必须可以被 8 整除。要确定可能的密钥大小，请使用 `cryptoadm list -vm` 命令。

- 创建密钥对并将其存储在 PKCS #11 密钥库中。

在使用此方法之前，必须先完成[步骤 1](#)。

PKCS #11 密钥库用于将对象存储在硬件设备上。该设备可以是插入到加密框架的 Sun Crypto Accelerator 6000 卡、受信任的平台模块 (trusted platform module, TPM) 设备或智能卡。PKCS #11 还可以用于将对象存储在 `softtoken`，或基于软件的令牌中，从而将这些对象存储在磁盘上的专用子目录中。有关更多信息，请参见 [pkcs11_softtoken\(5\)](#) 手册页。

可以从密钥库中按指定的标签检索密钥对。

```
% pktool genkeypair label=key-label \
[token=token[:manuf[:serial]]] \
[keytype=rsa|dsa|ec] [curve=ECC-Curve-Name]] \
[keylen=key-size] [listcurves]
```

label=key-label

为密钥对指定标签。可以从密钥库中按其标签检索密钥对。

`token=token[:manuf[:serial]]`

指定令牌名称。缺省情况下，令牌名称是 Sun Software PKCS#11 softtoken。

`keytype=rsa|dsa|ec [curve=ECC-Curve-Name]`

指定密钥对类型。对于椭圆曲线 (elliptic curve, ec) 类型，可选择性地指定曲线名称。曲线名称作为输出列出到 `listcurves` 选项。

`keylen=key-size`

指定密钥的长度（以位为单位）。该数字必须可以被 8 整除。

`listcurves`

列出可用作 ec 密钥类型的 `curve=` 选项值的椭圆曲线名称。

- 生成密钥对并将其存储在 NSS 密钥库中。

NSS 密钥库由依赖 NSS 作为其主加密接口的服务器使用。例如，Oracle iPlanet Web Server 将 NSS 数据库用于对象存储。

在使用此方法之前，必须先完成步骤 1。

```
% pktool keystore=nss genkeypair label=key-nickname \
[token=token[:manuf[:serial]]] \
[dir=directory-path] [prefix=database-prefix] \
[keytype=rsa|dsa|ec] [curve=ECC-Curve-Name] \
[keylen=key-size] [listcurves]
```

`keystore=nss`

值 `nss` 指定密钥存储位置的 NSS 类型。

`label=nickname`

为密钥对指定标签。可以从密钥库中按其标签检索密钥对。

`token=token[:manuf[:serial]]`

指定令牌名称。缺省情况下，令牌是 Sun Software PKCS#11 softtoken。

`dir=directory`

指定 NSS 数据库的目录路径。缺省情况下，`directory` 是当前目录。

`prefix=database-prefix`

指定 NSS 数据库的前缀。缺省为无前缀。

`keytype=rsa|dsa|ec [curve=ECC-Curve-Name]`

指定密钥对类型。对于椭圆曲线类型，可选择性地指定曲线名称。曲线名称作为输出列出到 `listcurves` 选项。

`keylen=key-size`

指定密钥的长度（以位为单位）。该数字必须可以被 8 整除。

`listcurves`

列出可用作 ec 密钥类型的 `curve=` 选项值的椭圆曲线名称。

3 可选检验密钥是否存在。

根据密钥的存储位置，使用以下命令之一：

- 检验 *key-filename* 文件中的密钥。

```
% pktool list keystore=file objtype=key infile=key-filename
Found n keys.
Key #1 - keytype:location (keylen)
```

- 检验 PKCS #11 密钥库中的密钥。

```
$ pktool list objtype=key
Enter PIN for keystore:
Found n keys.
Key #1 - keytype:location (keylen)
```

- 检验 NSS 密钥库中的密钥。

```
% pktool list keystore=nss dir=directory objtype=key
```

示例 13-6 使用 pktool 命令创建密钥对

在以下示例中，用户首次创建 PKCS #11 密钥库。确定 RSA 密钥对的密钥大小后，用户生成应用程序的密钥对。最后，用户验证该密钥对是否在密钥库中。用户注意到 RSA 密钥对的第二个实例可以存储在硬件上。由于用户未指定 *token* 参数，因此该密钥对存储为 Sun Software PKCS#11 softtoken。

```
# pktool setpin
Create new passphrase:      Easily remembered, hard-to-detect password
Re-enter new passphrase:    Retype password
Passphrase changed.
% cryptoadm list -vm | grep PAIR
...
CKM_DSA_KEY_PAIR_GEN        512  1024 . . .
CKM_RSA_PKCS_KEY_PAIR_GEN   256  4096 . . .
...
CKM_RSA_PKCS_KEY_PAIR_GEN   512  2048 X . .
ecc: CKM_EC_KEY_PAIR_GEN,CKM_ECDH1_DERIVE,CKM_ECDSA,CKM_ECDSA_SHA1
% pktool genkeypair label=specialappkeypair keytype=rsa keylen=2048
Enter PIN for Sun Software PKCS#11 softtoken :    Type password

% pktool list
Enter PIN for Sun Software PKCS#11 softtoken :    Type password

Found 1 keys.
Key #1 - keypair: specialappkeypair (2048 bits)
```

示例 13-7 创建使用椭圆曲线算法的密钥对

在以下示例中，用户将椭圆曲线 (elliptic curve, ec) 密钥对添加到密钥库中，指定曲线名称并验证该密钥对是否在密钥库中。

```
% pktool genkeypair listcurves
secp112r1, secp112r2, secp128r1, secp128r2, secp160k1
.
.
.
c2pnb304w1, c2tnb359v1, c2pnb368w1, c2tnb431r1, prime192v2
prime192v3
% pktool genkeypair label=eckeypair keytype=ec curves=c2tnb431r1
% pktool list
Enter PIN for Sun Software PKCS#11 softtoken : Type password

Found 2 keys.
Key #1 - keypair: specialappkeypair (2048 bits)
Key #2 - keypair: eckeypair (c2tnb431r1)
```

▼ 如何使用 `pktool signcsr` 命令签署证书请求

此过程用于签署 PKCS #10 证书签名请求 (certificate signing request, CSR)。CSR 可以是 PEM 或 DER 格式。签署过程颁发 X.509 v3 证书。要生成 PKCS #10 CSR，请参见 [pktool\(1\)](#) 手册页。

开始之前 您是证书颁发机构 (certificate authority, CA)，您已收到一个 CSR，该请求存储在文件中。

1 为 `pktool signcsr` 命令所需的参数收集以下信息：

signkey	如果已将签名者的密钥存储在 PKCS #11 密钥库中，则 signkey 是检索此私钥的 标签 。
	如果已将签名者的密钥存储在 NSS 密钥库或文件密钥库中，则 signkey 是保存此私钥的文件名。
csr	指定 CSR 的文件名。
serial	指定已签名证书的序列号。
outcer	指定已签名证书的文件名。
issuer	指定采用标识名 (distinguished name, DN) 格式的 CA 颁发者名称。

有关 `signcsr` 子命令的可选参数的信息，请参见 [pktool\(1\)](#) 手册页。

2 签署请求和颁发证书。

例如，以下命令可使用 PKCS #11 系统信息库中的签名者密钥签署证书。

```
# pktool signcsr signkey=CASigningKey \
csr=fromExampleCoCSR \
serial=0x12345678 \
outcert=ExampleCoCert2010 \
issuer="O=Oracle Corporation, \
OU=Oracle Solaris Security Technology, L=Redwood City, ST=CA, C=US, \
CN=rootsign Oracle"
```

以下命令使用文件中的签名者密钥签署证书。

```
# pktool signcsr signkey=CASigningKey \
csr=fromExampleCoCSR \
serial=0x12345678 \
outcert=ExampleCoCert2010 \
issuer="O=Oracle Corporation, \
OU=Oracle Solaris Security Technology, L=Redwood City, ST=CA, C=US, \
CN=rootsign Oracle"
```

3 将该证书发送给请求方。

您可以使用电子邮件、Web 站点或其他机制将该证书传送给请求方。

例如，您可以使用电子邮件将 ExampleCoCert2010 文件发送给请求方。

▼ 如何在 KMF 中管理第三方插件

通过向插件提供密钥库名称来标识该插件。将插件添加到 KMF 中时，软件会按其密钥库名称标识它。可以定义该插件以接受选项。此过程包括如何从 KMF 中删除插件。

1 安装插件。

```
% /usr/bin/kmfcfg install keystore=keystore-name \
modulepath=path-to-plugin [option="option-string"]
```

其中

keystore-name—指定您提供的密钥库的唯一名称。

path-to-plugin—指定 KMF 插件的共享库对象的完整路径。

option-string—指定共享库对象的可选参数。

2 列出插件。

```
% kmfcfg list plugin
keystore-name:path-to-plugin [(built-in)] | [;option=option-string]
```

3 要删除插件，请先卸载该插件，然后检验是否已删除。

```
% kmfcfg uninstall keystore=keystore-name
% kmfcfg plugin list
```

示例 13-8 使用选项调用 KMF 插件

在以下示例中，管理员将 KMF 插件存储在站点专用目录中。定义该插件以接受 debug 选项。管理员添加插件，然后验证该插件是否已安装。

```
# /usr/bin/kmfcfg install keystore=mykmfplug \  
modulepath=/lib/security/site-modules/mykmfplug.so  
# kmfcfg list plugin  
KMF plugin information:  
-----  
pkcs11:kmf_pkcs11.so.1 (built-in)  
file:kmf_openssl.so.1 (built-in)  
nss:kmf_nss.so.1 (built-in)  
mykmfplug:/lib/security/site-modules/mykmfplug.so  
# kmfcfg modify plugin keystore=mykmfplug option="debug"  
# kmfcfg list plugin  
KMF plugin information:  
-----  
...  
mykmfplug:/lib/security/site-modules/mykmfplug.so;option=debug
```

该插件现在在调试模式下运行。

第 5 部分

验证服务和安全通信

本部分介绍可以在非联网系统上或两个系统之间配置的验证服务。

- 第 14 章，网络服务验证（任务）
- 第 15 章，使用 PAM
- 第 16 章，使用 SASL
- 第 17 章，使用 安全 Shell（任务）
- 第 18 章，安全 Shell（参考）

要配置经过验证的用户和系统组成的网络，请参见第 6 部分。

网络服务验证（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 247 页中的“安全 RPC 概述”
- 第 251 页中的“使用安全 RPC 管理验证（任务）”

安全 RPC 概述

安全 RPC（Remote Procedure Call，远程过程调用）通过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 NIS 命名服务。

NFS 服务和安全 RPC

NFS 支持多个主机通过网络共享文件。在 NFS 服务中，一个服务器可为多个客户机保存数据和资源。这些客户机具有服务器与客户机共享文件系统的访问权限。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这些文件系统。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 的最常见用途之一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 加密以确保授权访问。AUTH_DH 机制也称为 AUTH_DES。有关更多信息，请参见以下内容：

- 要设置和管理安全 NFS，请参见《[Oracle Solaris 管理：网络服务](#)》中的“[管理安全 NFS 系统](#)”。
- 有关涉及 RPC 验证的事务的概述，请参见第 248 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 位密钥进行数据加密。如果两个凭证用户或主体知道同一 DES 密钥，他们就可以通过使用此密钥加密和解密文本进行秘密通信。DES 是一种相对快速的加密机制。

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密文消息，从而能够发现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 开发的一种验证系统。Kerberos 中的部分加密基于 DES。安全 RPC 中不再提供 Kerberos V4 支持。但是，此发行版中包括使用 RPCSEC_GSS 的 Kerberos V5 客户端和服务端实现。有关更多信息，请参见第 19 章，[Kerberos 服务介绍](#)。

Diffie-Hellman 验证和安全 RPC

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，它们将其与公钥一起使用以设计公用密钥。私钥也称为**密钥** (secret key)。客户机和服务器使用公用密钥相互通信。公用密钥使用公认的加密功能（例如 DES）进行加密。

验证成功的基础是发送系统能够使用公用密钥来加密当前时间。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《[Oracle Solaris 管理：网络服务](#)》中的“[管理网络时间协议（任务）](#)”。

公钥和私钥存储在 NIS 数据库中。NIS 将密钥存储在 publickey 映射中。此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 映射以及为每个用户生成公钥和私钥。私钥以加密形式与用户口令存储在一起。此过程使私钥只让用户知道。

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) 的事务系列。

生成安全 RPC 的公钥和密钥

执行事务之前，管理员有时会运行 `newkey` 或 `nisaddcred` 命令来生成公钥和密钥。每个用户都有唯一的公钥和密钥。公钥存储在公共数据库中。密钥以加密形式存储在同一个数据库中。`chkey` 命令可更改密钥对。

运行安全 RPC 的 `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，用户就必须登录，然后运行 `keylogin` 命令。

`keylogin` 命令会提示用户提供安全 RPC 口令。然后，此命令将使用该口令对密钥进行解密。随后，`keylogin` 命令将解密的密钥传递给 `keyserver` 程序。`keyserver` 是一个在每台计算机上都有一个本地实例的 RPC 服务。`keyserver` 会保存解密的密钥，并等待用户在服务器上启动安全 RPC 事务。

如果登录口令与 RPC 口令相同，则登录过程会将密钥传递给 `keyserver`。如果要求不同的口令，则用户在任何时候都必须运行 `keylogin` 命令。如果 `keylogin` 命令包括在用户的环境配置文件（如 `~/.login`、`~/.cshrc` 或 `~/.profile` 文件）中，则用户登录时会自动运行 `keylogin` 命令。

生成安全 RPC 的对话密钥

用户在某个服务器上启动事务时，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戳进行加密。
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创建一个公用密钥。
5. `keyserver` 使用此公用密钥来加密此对话密钥。

初次与安全 RPC 中的服务器通信

然后，会将包含加密的时间戳和对话密钥的传输内容发送到服务器。传输内容包括凭证和检验器。凭证包含三个组成部分：

- 客户机的网络名称
- 使用公用密钥加密后的对话密钥
- 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戳之间存在的时间差异。如果服务器时钟与时间戳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：

- 加密的时间戳
- 按 1 递减的指定窗口的已加密检验器

当某人需要模拟用户时，需要使用窗口检验器。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成某个随机密钥。然后，服务器使用此密钥尝试对窗口和时间戳进行解密。结果是随机数字。但是，经过几千次尝试之后，随机窗口/时间戳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。

在安全 RPC 模式下解密对话密钥

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，因此计算过程需要知道其中一个密钥。
3. 内核使用公用密钥来解密此对话密钥。
4. 内核调用 `keyserver` 以使用解密的对话密钥对客户机的时间戳进行解密。

在安全 RPC 模式下在服务器上存储信息

服务器对客户机的时间戳进行解密之后，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
- 客户机的时间戳

服务器存储前三项供将来使用，服务器存储客户机的时间戳以防止重放。服务器只接受时间上晚于最新时间戳的时间戳。因此，可保证拒绝任何重放的事务。

注 - 调用者的名称将隐含在这些事务中，必须通过某种方式验证此调用方。`keyserver` 不能使用 DES 验证来验证调用者，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。

在安全 RPC 模式下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 的客户机时间戳（由对话密钥加密）

从客户机的时间戳中减 1 的原因是确保此时间戳过时。过时的时间戳不能再用作客户机检验器。

在安全 RPC 模式下验证服务器

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戳。

在安全 RPC 模式下处理事务

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。客户机还会发送另一个加密的时间戳。服务器会将使用对话密钥加密的客户机时间戳减 1 值发送回来。

使用安全 RPC 管理验证（任务）

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。

管理安全 RPC（任务列表）

以下任务列表列出了为 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 251 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 252 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 253 页中的“如何为 NIS 用户设置 Diffie-Hellman 密钥”
4. 通过验证共享 NFS 文件。	使 NFS 服务器可以使用验证安全地保护共享的文件系统。	第 254 页中的“如何通过 Diffie-Hellman 验证共享 NFS 文件”

▼ 如何重新启动安全 RPC Keyserver

开始之前 您必须是 root 角色。

1 验证 keyserv 守护进程是否正在运行。

```
# svcs \*keyserver\*
STATE      STIME      FMRI
disabled   Dec_14    svc:/network/rpc/keyserver
```

- 2 如果 **keyserver** 服务未联机，则启用该服务。

```
# 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。

开始之前 您必须是 **root** 角色。

- 1 如果缺省命名服务不是 NIS，则将 **publickey** 映射添加到命名服务。

- a. 确认命名服务的 **config/default** 值不是 **nis**。

```
# svccfg -s name-service/switch listprop config
config                                application
config/value authorization            astring      solaris.smf.value.name-service.switch
config/default                       astring      files
config/host                          astring      "files nis dns"
config/printer                       astring      "user files nis"
```

如果 **config/default** 值为 **nis**，您可以在此处停止。

- b. 将命名服务的 **publickey** 设置为 **nis**。

```
# svccfg
# svccfg -s name-service/switch setprop config/publickey = astring: "nis"
# svccfg -s name-service/switch:default refresh
```

- c. 确认 **publickey** 值。

```
# svccfg
# svccfg -s name-service/switch listprop
config                                application
config/value authorization            astring      solaris.smf.value.name-service.switch
config/default                       astring      files
config/host                          astring      "files nis dns"
config/printer                       astring      "user files nis"
config/publickey                     astring      nis
```

在此系统上，列出 **publickey** 的值，因为它不同于缺省值 **files**。

- 2 使用 **newkey** 命令创建一个新密钥对。

```
# newkey -h hostname
```

其中，*hostname* 是客户机的名称。

示例 14-1 在 NIS 客户机上为 root 设置新密钥

在以下示例中，将 **earth** 设置成了安全 NIS 客户机。为管理员分配了 "Name Service Security"（名称服务安全）权限配置文件。


```
# newkey -h earth
Adding new key for unix.earth@example.com
New Password:      <Type password>
Retype password:   <Retype password>
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 只有已登录到 NIS 主服务器的系统管理员才能为用户生成新密钥。必须为管理员分配 "Name Service Security"（名称服务安全）权限配置文件。

- 1 成为具有所需安全属性的管理员。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。

- 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。私钥是使用此通用口令以加密形式存储的。

- 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本人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 14-2 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:      <Type password>
Retype password:   <Retype password>
Please wait for the database to get updated...
Your new key has been successfully stored away.
#
```

然后，用户 **jdoe** 使用专用口令对该密钥重新加密。

```
% chkey -p
Updating nis publickey database.
Reencrypting key for unix.12345@example.com
```

```
Please enter the Secure-RPC password for jdoe:    <Type password>
Please enter the login password for jdoe:         <Type password>
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。要启用网络验证，请完成第 252 页中的“如何为 NIS 主机设置 Diffie-Hellman 密钥”中的步骤。

您必须分配有 System Management（系统管理）权限配置文件，才能执行此任务。

- 1 成为具有所需安全属性的管理员。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。

- 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* 是共享的文件系统。

-o sec=dh 选项表示现在需要通过 AUTH_DH 验证后才能访问文件系统。

- 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，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。

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 Oracle Solaris OS 中“插入”验证服务的方法。PAM 可以为访问系统时的多项验证服务提供支持。

- 第 255 页中的“PAM（概述）”
- 第 257 页中的“PAM（任务）”
- 第 259 页中的“PAM 配置（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（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
 - 基于每个应用程序的验证策略
 - 可以选择缺省验证机制
 - 能够在高安全性系统中要求提供多个口令
- 易于最终用户使用
 - 如果不同的验证服务使用相同口令，无需重新键入口令
 - 可以提示用户为多个验证服务输入口令，而不要求用户键入多个命令
- 可以将可选选项传递到用户验证服务

- 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 框架介绍

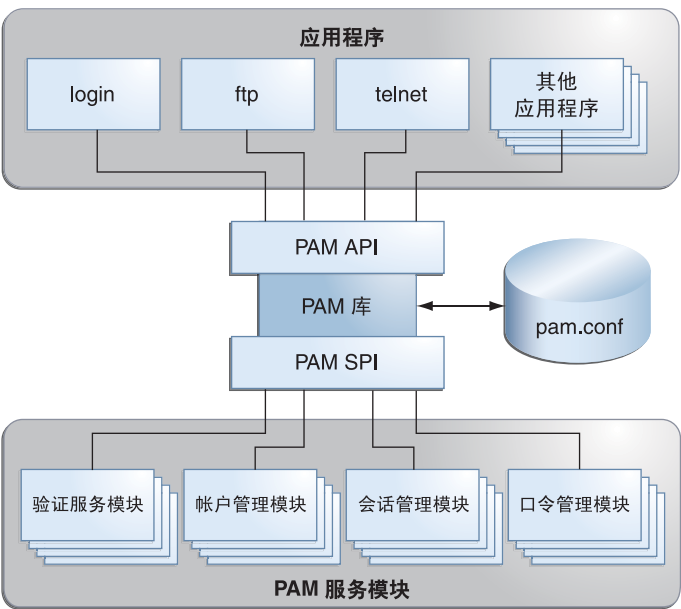
PAM 框架由四个部分组成：

- PAM 使用者
- PAM 库
- `pam.conf(4)` 配置文件
- PAM 服务模块，也称为提供者

该框架为与验证相关的活动提供了统一的执行方式。采用该方式，应用程序开发者不必了解策略的语义即可使用 PAM 服务。算法是集中提供的。可以独立于各个应用程序对算法进行修改。借助 PAM，管理员可以根据特定系统的需要调整验证过程，而不必更改任何应用程序。这种调整是通过 PAM 配置文件 `pam.conf` 进行的。

下图说明了 PAM 体系结构。应用程序通过 PAM 应用编程接口 (application programming interface, API) 与 PAM 库进行通信。PAM 模块通过 PAM 服务提供者接口 (service provider interface, SPI) 与 PAM 库进行通信。通过这种方式，PAM 库可使应用程序和模块相互进行通信。

图 15-1 PAM 体系结构



此发行版中 PAM 的更改

Oracle Solaris 11 Express 发行版的 PAM 框架包括新的 `pam_allow` 模块。此模块可以用于向所有用户授权访问权限，而无需实施任何安全性。使用此模块时需谨慎。有关更多信息，请参见 [pam_allow\(5\)](#) 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 257 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 257 页中的“规划 PAM 实现”
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 258 页中的“如何添加 PAM 模块”
阻止访问 <code>~/.rhosts</code> 。	通过阻止访问 <code>~/.rhosts</code> 来进一步提高安全性。	第 259 页中的“如何使用 PAM 防止从远程系统进行 Rhost 样式的访问”
启动错误日志记录。	通过 <code>syslog</code> 启动 PAM 错误消息日志记录。	第 259 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，则应关注以下问题：

- 确定您的需求，特别是应选择的 PAM 服务模块。
- 确定需要特殊配置选项的服务。使用 `other`（如果适用）。
- 决定模块的运行顺序。
- 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 260 页中的“PAM 堆栈工作原理”。
- 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。
- 查阅与模块关联的手册页。这些手册页可帮助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，则任何用户都可能无法登录。由于 `sulogin` 命令不使用 PAM，因此，需要使用超级用户口令将计算机引导至单用户模式并修复问题。

更改了 `/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` 命令并检验所做的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 **成为管理员。**
有关更多信息，请参见第 150 页中的“如何获取管理权限”。
- 2 **确定应使用的控制标志和其他选项。**
有关控制标志的信息，请参见第 260 页中的“PAM 堆栈工作原理”。
- 3 **确保设置了所有权和权限，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该模块。**

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 `ssh`）登录，然后运行 `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 成为管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 从 PAM 配置文件中删除所有包含 `rhhosts_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` 访问都要求提供口令。

3 禁用 `rsh` 服务。

要防止对 `~/.rhosts` 文件进行其他未经验证的访问，请记住禁用 `rsh` 服务。

```
# svcadm disable network/shell
```

▼ 如何记录 PAM 错误报告

1 成为管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 配置 `/etc/syslog.conf` 文件以设置您需要的日志记录级别。

有关日志记录级别的更多信息，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 (参考)

PAM 配置文件 `pam.conf(4)` 用于为系统服务（如 `login`、`rlogin`、`su` 和 `cron`）配置 PAM 服务模块。系统管理员可以管理此文件。如果 `pam.conf` 中的项顺序有误，会导致无法预料的负面影响。例如，如果 `pam.conf` 配置错误，则会将多个用户锁定在外，这样必须采用单用户模式才能进行修复。有关设置顺序的说明，请参见第 260 页中的“PAM 堆栈工作原理”。

PAM 配置文件语法

该配置文件中的项采用以下格式：

```
service-name module-type control-flag module-path module-options
```

<i>service-name</i>	服务的名称，例如 ftp、login 或 passwd。应用程序可以针对其提供的各服务使用不同的服务名。例如，Oracle Solaris 安全 shell 守护进程使用以下服务名：sshd-none、sshd-password、sshd-kbdint、sshd-pubkey 以及 sshd-hostbased。服务名 <i>other</i> 是用作通配符服务名的预定义名称。如果在配置文件中未找到特定的服务名，则会使用 <i>other</i> 的配置。
<i>module-type</i>	服务类型，即 auth、account、session 或 password。
<i>control-flag</i>	用于指明在确定服务的集成成败值过程中模块所起的作用。有效的控制标志包括 binding、include、optional、required、requisite 和 sufficient。有关这些标志用法的信息，请参见第 260 页中的“PAM 堆栈工作原理”。
<i>module-path</i>	用于实现服务的库对象的路径。如果路径名不是绝对路径名，则认为路径名相对于 /usr/lib/security/\$ISA/。通过使用与体系结构有关的宏 \$ISA，libpam 可查看应用程序特定体系结构的目录。
<i>module-options</i>	传递给服务模块的选项。模块的手册页介绍了相应模块可接受的选项。典型的模块选项包括 nowarn 和 debug。

PAM 堆栈工作原理

应用程序调用以下函数时，libpam 将读取配置文件 /etc/pam.conf 以确定针对该服务参与操作的模块：

- pam_authenticate(3PAM)
- pam_acct_mgmt(3PAM)
- pam_setcred(3PAM)
- pam_open_session(3PAM)
- pam_close_session(3PAM)
- pam_chauthtok(3PAM)

如果对于该服务的操作（如验证或帐户管理），/etc/pam.conf 仅包含一个模块，则该模块的结果将确定操作的结果。例如，passwd 应用程序的缺省验证操作包含一个模块 pam_passwd_auth.so.1：

```
passwd auth required pam_passwd_auth.so.1
```

另一方面，如果为服务操作定义了多个模块，那么这些模块就会堆叠起来，即，对于该服务存在一个 PAM 堆栈。例如，请考虑 pam.conf 包含以下项的情况：

```
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
```

这些项表示 `login` 服务的 `auth` 栈样例。要确定此栈的结果，各个模块的结果代码需要进行**集成处理**。在集成处理过程中，模块按照 `/etc/pam.conf` 中指定的顺序执行。每个成功/失败代码都会根据模块的控制标志集成到整体结果中。控制标志可能会导致栈很早就终止。例如，`requisite` 模块可能会失败，而 `sufficient` 或 `binding` 模块则可能会成功。处理栈之后，各个结果会组合成一个整体结果，传送给应用程序。

控制标志用于指明 PAM 模块在确定对服务的访问方面所起的作用。控制标志及其作用如下所述：

- **Binding**—如果前面必需的模块都没有失败，则成功满足绑定模块的要求后，会向应用程序立即返回成功信息。如果满足这些条件，则不会进一步执行模块。如果失败，则会记录必需的失败信息并继续处理模块。
- **Include**—添加单独的 PAM 配置文件中的行，以便在 PAM 堆栈中的此点使用。此标志不控制成功或失败行为。读取一个新文件时，PAM `include` 栈值会递增。当新文件中的栈检查完成时，`include` 栈值会递减。如果到达文件末尾，并且 PAM `include` 栈值为 0，则栈处理结束。PAM `include` 栈的最大数为 32。
- **Optional**—不必成功满足可选模块的要求即可使用服务。如果失败，则会记录可选的失败信息。
- **Required**—必须成功满足必需模块的要求才能使用服务。如果失败，则执行该服务的其余模块后会返回错误信息。仅当绑定模块或必需模块均没有报告失败的情况下，才会返回该服务最终成功的信息。
- **Requisite**—必须成功满足必备模块的要求才能使用服务。如果失败，则会立即返回错误信息，而不会进一步执行模块。只有一个服务的所有必备模块都返回成功信息，函数才能向应用程序返回成功信息。
- **Sufficient**—如果前面的必需模块的要求都成功满足，则成功满足控制标志为 `sufficient` 的模块的要求后将向应用程序立即返回成功信息，而不会进一步执行模块。如果失败，则会记录可选的失败信息。

下面的两个图说明了如何在集成处理过程中确定访问权限。第一个图说明如何为每种类型的控制标志记录成败信息。第二个图说明如何确定集成值。

图 15-2 PAM 堆栈：控制标志的作用

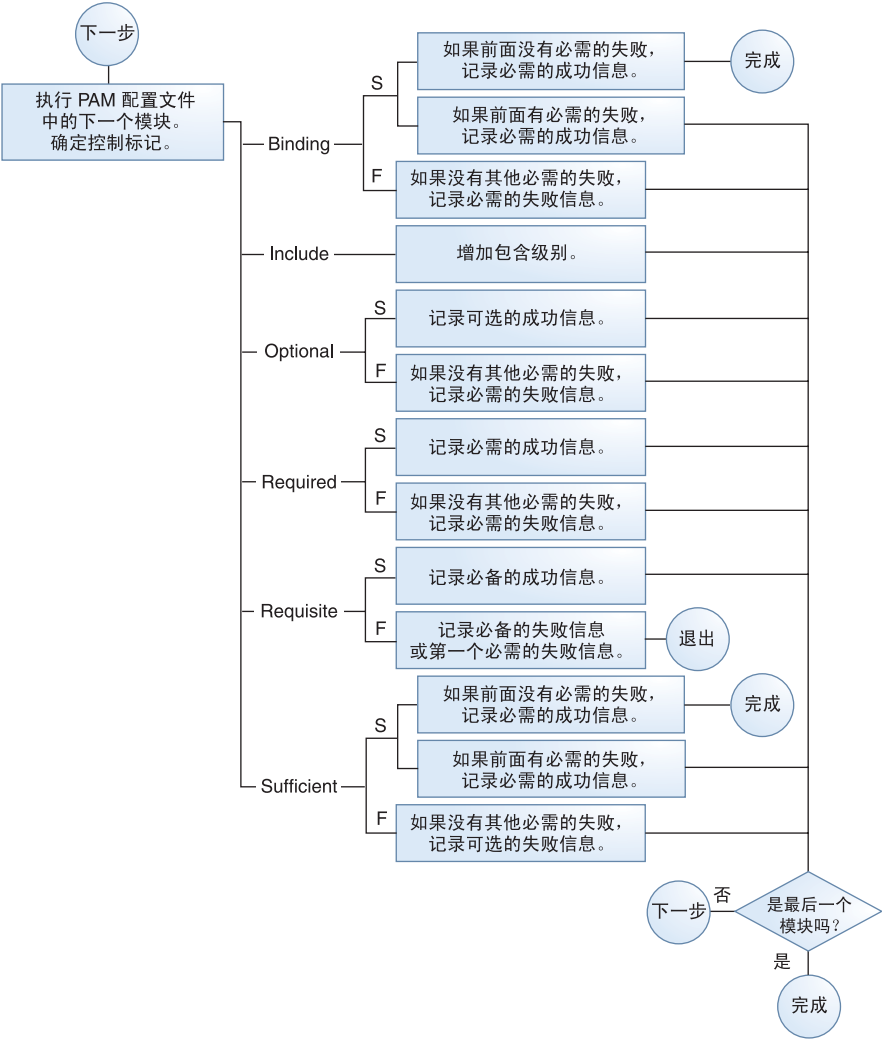
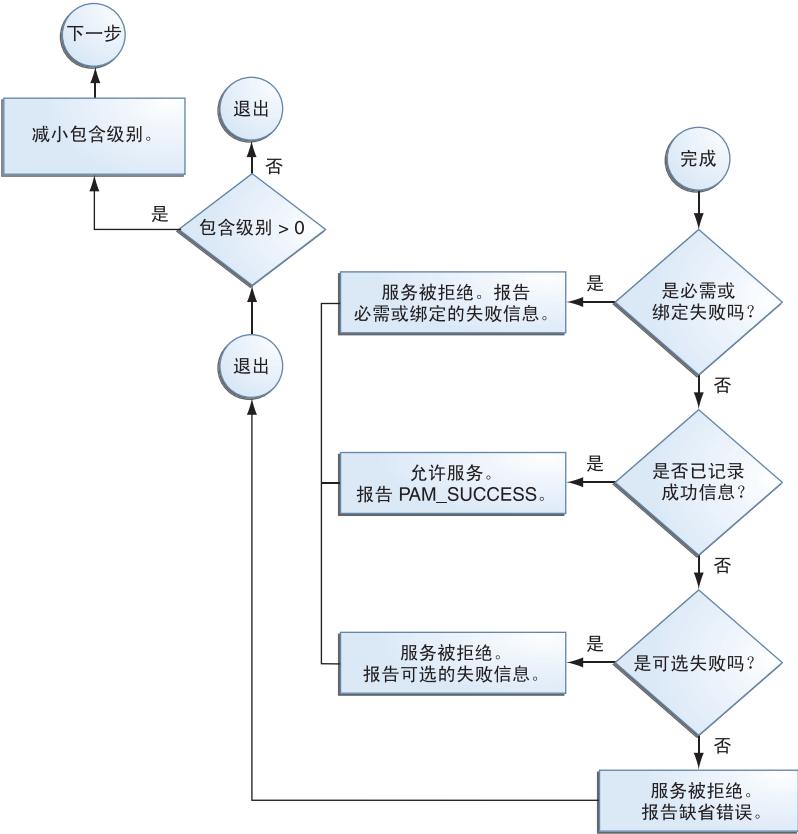


图 15-3 PAM 堆栈：如何确定集成值



PAM 堆栈示例

请考虑以下请求验证的 rlogin 服务示例。

示例 15-1 典型 PAM 配置文件的部分内容

该示例中的 pam.conf 文件包含以下有关 rlogin 服务的内容：

```
# Authentication management
...
# rlogin service
rlogin auth sufficient pam_rhosts_auth.so.1
rlogin auth requisite pam_authok_get.so.1
rlogin auth required pam_dhkeys.so.1
rlogin auth required pam_unix_auth.so.1
...
```

示例 15-1 典型 PAM 配置文件的部分内容 (续)

当 `rlogin` 服务请求验证时，`libpam` 会首先执行 `pam_rhosts_auth(5)` 模块。对于 `pam_rhosts_auth` 模块，控制标志将设置为 `sufficient`。如果 `pam_rhosts_auth` 模块可以验证用户，则处理过程将停止，并且会向应用程序返回成功信息。

如果 `pam_rhosts_auth` 模块无法验证用户，则将执行下一个 PAM 模块 `pam_authtok_get(5)`。此模块的控制标志将设置为 `requisite`。如果 `pam_authtok_get` 失败，则验证过程将结束并向 `rlogin` 返回失败信息。

如果 `pam_authtok_get` 成功，则将执行接下来的两个模块 `pam_dhkeys(5)` 和 `pam_unix_auth(5)`。这两个模块的关联控制标志都会设置为 `required`，以便处理过程可继续进行，无论是否会返回单独的失败信息。执行了 `pam_unix_auth` 后，不会再保留任何用于 `rlogin` 验证的模块。此时，如果 `pam_dhkeys` 或 `pam_unix_auth` 返回失败信息，则会拒绝用户通过 `rlogin` 进行访问。

使用 SASL

本章介绍有关简单验证和安全层 (Simple Authentication and Security Layer, SASL) 的信息。

- [第 265 页中的“SASL（概述）”](#)
- [第 265 页中的“SASL（参考）”](#)

SASL（概述）

简单验证和安全层 (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`，此库提供该应用程序与各种 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。此 SASL 版本源于 Cyrus SASL，但进行了几处更改。

SASL 提供以下服务：

- 装入任何插件
- 确定应用程序帮助选择安全机制时必需的安全选项
- 列出应用程序可用的插件
- 为特定验证从可用机制列表中选择最佳机制
- 在应用程序与所选机制之间路由验证数据
- 将有关 SASL 协商的信息返回给应用程序

SASL（参考）

本节提供有关 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。提供了以下安全机制插件：

<code>crammd5.so.1</code>	CRAM-MD5，仅支持验证，不支持授权。
<code>digestmd5.so.1</code>	DIGEST-MD5，支持验证、完整性、保密性和授权。
<code>gssapi.so.1</code>	GSSAPI，支持验证、完整性、保密性和授权。GSSAPI 安全机制需要使用有效的 Kerberos 基础结构。
<code>plain.so.1</code>	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，Oracle Solaris 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发挥完全功能，用户必须提供 `auxprop` 插件以检索明文口令。PLAIN 插件还需要额外支持才能验证口令。对口令验证的支持包括：对服务器应用程序的回调、`auxprop` 插件、`saslauthd` 或 `pwcheck`。Oracle Solaris 发行版中未提供 `saslauthd` 和 `pwcheck` 守护进程。要实现更好的互操作性，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机制。

SASL 环境变量

缺省情况下，客户机验证名称设置为 `getenv("LOGNAME")`。客户机或插件可以重置此变量。

SASL 选项

通过使用可在 `/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。

支持以下选项：

<code>auto_transition</code>	用户成功进行纯文本验证后自动将该用户转换到其他机制。
<code>auxprop_login</code>	列出要使用的辅助属性插件的名称。
<code>canon_user_plugin</code>	选择要使用的 <code>canon_user</code> 插件。
<code>mech_list</code>	列出允许服务器应用程序使用的机制。

<code>pwcheck_method</code>	列出用于验证口令的机制。目前， <code>auxprop</code> 是唯一允许的值。
<code>reauth_timeout</code>	设置对验证信息进行高速缓存以便进行快速重新验证的时间（以分钟为单位）。此选项由 <code>DIGEST-MD5</code> 插件使用。将此选项设置为 <code>0</code> 将禁用重新验证。

不支持以下选项：

<code>plugin_list</code>	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。
<code>saslauthd_path</code>	定义用于与 <code>saslauthd</code> 守护进程通信的 <code>saslauthd</code> 门的位置。Oracle Solaris 发行版中没有包括 <code>saslauthd</code> 守护进程。因此，也不包括此选项。
<code>keytab</code>	定义 GSSAPI 插件使用的 <code>keytab</code> 文件的位置。可使用 <code>KRB5_KTNAME</code> 环境变量而不是此选项来设置缺省的 <code>keytab</code> 位置。

Cyrus SASL 中不包括以下选项。但是，Oracle Solaris 发行版中添加了这些选项：

<code>use_authid</code>	创建 GSS 客户机安全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。
<code>log_level</code>	为服务器设置所需的日志记录级别。

使用 安全 Shell（任务）

使用 Oracle Solaris 的安全 Shell 功能，可以通过不安全的网络对远程主机进行安全访问。该 shell 提供了用于远程登录和远程文件传输的命令。以下是本章中的主题列表。

- 第 269 页中的“安全 Shell（概述）”
- 第 271 页中的“安全 Shell 和 OpenSSH 项目”
- 第 272 页中的“安全 Shell 和 FIPS-140 支持”
- 第 272 页中的“安全 Shell（任务列表）”

有关参考信息，请参见第 18 章，安全 Shell（参考）。

安全 Shell（概述）

在安全 Shell 中，通过使用口令和/或公钥提供验证。所有网络通信都会被加密。因此，安全 Shell 可防止潜在入侵者读取被拦截的通信。安全 Shell 还可防止入侵者欺骗系统。

安全 Shell 还可以用作即时请求 [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与远程计算机之间的各个端口号。

使用安全 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。

在服务器端，安全 Shell 支持两个版本的安全 Shell 协议，版本 1 (v1) 和版本 2。版本 2 (v2) 更为安全。安全 Shell 提供 v1 仅是为帮助正在迁移至 v2 的用户。有关 v1 的信息，请参见《[System Administration Guide: Security Services](#)》。

安全 Shell 验证

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，因为从不通过网络传送私钥。

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《[Oracle Solaris 11 开发者安全性指南](#)》中的“[GSS-API 介绍](#)”。
- **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**—通过用户的 RSA 和 DSA 公钥/私钥来验证用户。
- **口令验证**—使用 PAM 来验证用户。v2 中的键盘验证方法允许 PAM 进行任意提示。有关更多信息，请参见[sshd\(1M\)](#)手册页中的 SECURITY 部分。

下表列出了验证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。该表中列出了安全 Shell 验证方法、兼容的协议版本和主机要求。

表 17-1 安全 Shell 的验证方法

验证方法	本地主机（客户机）要求	远程主机（服务器）要求
GSS-API	GSS 机制的启动器凭证。	GSS 机制的接受器凭证。有关更多信息，请参见第 288 页中的“在安全 Shell 中获取 GSS 凭证”。
基于主机	用户帐户	用户帐户
	<code>/etc/ssh/ssh_host_rsa_key</code> 或 <code>/etc/ssh/ssh_host_dsa_key</code> 中的本地主机私钥	<code>/etc/ssh/known_hosts</code> 或 <code>~/.ssh/known_hosts</code> 中的本地主机公钥
	<code>/etc/ssh/ssh_config</code> 中的 <code>HostbasedAuthentication yes</code>	<code>/etc/ssh/sshd_config</code> 中的 <code>HostbasedAuthentication yes</code>
		<code>/etc/ssh/sshd_config</code> 中的 <code>IgnoreRhosts no</code> <code>/etc/ssh/shosts.equiv</code> 、 <code>/etc/hosts.equiv</code> 、 <code>~/.rhosts</code> 或 <code>~/.shosts</code> 中的本地主机项
RSA 或 DSA 公钥	用户帐户	用户帐户
	<code>~/.ssh/id_rsa</code> 或 <code>~/.ssh/id_dsa</code> 中的私钥 <code>~/.ssh/id_rsa.pub</code> 或 <code>~/.ssh/id_dsa.pub</code> 中的 用户公钥	<code>~/.ssh/authorized_keys</code> 中的用户公钥
基于口令	用户帐户	用户帐户
		支持 PAM。

表 17-1 安全 Shell 的验证方法 (续)

验证方法	本地主机 (客户机) 要求	远程主机 (服务器) 要求
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/ssh/shosts.equiv、/etc/hosts.equiv、 ~/.shosts 或 ~/.rhosts 中的本地主机项

企业中的安全 Shell

有关 Oracle Solaris 系统中的安全 Shell 的全面讨论，请参见 Jason Reid 所著的《Secure Shell in the Enterprise》，2003 年 6 月出版，ISBN 为 0-13-142900-0。本书是 Sun Microsystems Press 出版的 Sun BluePrints 系列的一部分。

安全 Shell 和 OpenSSH 项目

安全 Shell 是 [OpenSSH \(http://www.openssh.com\)](http://www.openssh.com) 项目的一个分支。安全 Shell 中集成了在较新 OpenSSH 版本中发现的漏洞的安全修复，就像是单独的错误修复和功能一样。关于安全 Shell 分支的内部开发仍在继续。

在此安全 Shell 发行版中，针对 v2 协议实施以下功能：

- ForceCommand 关键字—强制执行指定的命令，无论用户在命令行输入什么内容。此关键字在 Match 数据块中非常有用。此 sshd_config 配置选项类似于 \$HOME/.ssh/authorized_keys 中的 command="..." 选项。
- AES-128 口令短语保护—在本发行版中，对 ssh-keygen 命令生成的私钥使用 AES-128 算法进行保护。此算法保护新生成的密钥和重新加密的密钥，如口令短语更改时的密钥。
- sftp-server 命令的 -u 选项—允许用户对文件和目录设置明确的 umask。此选项可覆盖用户的缺省 umask。有关示例，请参见 sshd_config(4) 手册页中对 Subsystem 的说明。
- Match 块的其他关键字—AuthorizedKeysFile、ForceCommand 和 HostbasedUsesNameFromPacketOnly 在 Match 块内受支持。缺省情况下，AuthorizedKeysFile 的值为 \$HOME/.ssh/authorized_keys，HostbasedUsesNameFromPacketOnly 为 no。要使用 Match 块，请参见第 276 页中的“如何创建不同于 SSH 系统缺省值的用户和主机例外”。

Oracle Solaris 工程师不仅为该项目提供错误修复，他们还将以下 Oracle Solaris 功能集成到安全 Shell 分支中：

- PAM—安全 Shell 使用 PAM。不支持 OpenSSH UsePAM 配置选项。
- 特权分离—安全 Shell 不使用来自于 OpenSSH 项目的特权分离代码。安全 Shell 将审计处理、记录保存和重新生成密钥与会话协议的处理相分离。
安全 Shell 特权分离代码始终开启，无法关闭。不支持 OpenSSH UsePrivilegeSeparation 选项。
- 语言环境—安全 Shell 完全支持 RFC 4253 《Secure Shell Transfer Protocol》中定义的语言协商。用户登录后，该用户的登录 shell 配置文件可以覆盖安全 Shell 协商的语言环境设置。
- 审计—安全 Shell 完全集成到 Solaris 审计服务中。有关审计服务的详细信息，请参见第 7 部分。
- GSS-API 支持—GSS-API 可用于用户验证和初始密钥交换。GSS-API 在 RFC4462 《Generic Security Service Application Program Interface》中进行定义。
- 代理命令—安全 Shell 为 SOCKS5 和 HTTP 协议提供了代理命令。有关示例，请参见第 283 页中的“如何设置到防火墙外部主机的缺省连接”。

在 Oracle Solaris 发行版中，安全 Shell 从 OpenSSH 项目中重新同步 SSH_OLD_FORWARD_ADDR 兼容性标志。截止 2011 年 3 月，安全 Shell 版本为 1.5。

安全 Shell 和 FIPS-140 支持

使用 Sun Crypto Accelerator 6000 卡进行安全 Shell 操作时，安全 Shell 在第 3 级 FIPS-140 支持下运行。第 3 级硬件经过认证，可抵御物理篡改，使用基于身份的验证，并将处理关键安全参数的接口同硬件的其他接口隔离。

安全 Shell（任务列表）

以下任务列表链接到有关配置安全 Shell 和使用 Oracle Solaris 中的安全 Shell 功能的任务列表。

任务	说明	参考
配置安全 Shell。	指导管理员为用户配置安全 Shell。	第 273 页中的“配置安全 Shell（任务列表）”
使用安全 Shell。	指导用户使用安全 Shell。	第 277 页中的“使用安全 Shell（任务列表）”

配置安全 Shell（任务）

缺省情况下，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，要使端口转发正常工作，也需要管理干预。

配置安全 Shell（任务列表）

以下任务列表列出了有关配置安全 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 273 页中的“如何为安全 Shell 设置基于主机的验证”
配置端口转发。	允许用户使用端口转发。	第 275 页中的“如何在安全 Shell 中配置端口转发”
配置 SSH 系统缺省值例外。	对于用户、主机、组和地址，指定与系统缺省值不同的 SSH 设置。	第 276 页中的“如何创建不同于 SSH 系统缺省值的用户和主机例外”

▼ 如何为安全 Shell 设置基于主机的验证

以下过程设置一个公钥系统，其中使用客户机的公钥进行服务器上的验证。用户还必须创建一个公钥/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**是指客户机尝试访问的计算机。

开始之前 您必须是 root 角色。

- 1

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`
有关该文件的语法，请参见 [ssh_config\(4\)](#) 手册页
- 2

在服务器上，启用基于主机的验证。

在服务器验证文件 `/etc/ssh/sshd_config` 中，键入相同的项：
`HostbasedAuthentication yes`
有关该文件的语法，请参见 [sshd_config\(4\)](#) 手册页

- 3 在服务器上，配置允许将客户机识别为可信主机的文件。

有关更多信息，请参见 [sshd\(1M\)](#) 手册页的 FILES 部分。

- 将客户机作为一项添加到服务器的 `/etc/ssh/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

- 4 在服务器上，确保 `sshd` 守护进程可以访问可信主机列表。

在 `/etc/ssh/sshd_config` 文件中，将 `IgnoreRhosts` 设置为 `no`。

```
## sshd_config
IgnoreRhosts no
```

- 5 确保站点上的安全 Shell 用户在两台主机上都拥有帐户。

- 6 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 277 页中的“如何生成用于安全 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，在一行中键入以下命令（不带反斜杠）。

```
# cat /etc/ssh/ssh_host_dsa_key.pub | ssh RemoteHost \
'cat >> /etc/ssh/ssh_known_hosts && echo "Host key copied"'
```

- b. 出现提示时，提供登录口令。

复制该文件后，将显示 "Host key copied"（主机密钥已复制）消息。

`/etc/ssh/ssh_known_hosts` 文件中的每一行均由空格分隔的字段组成：

hostnames algorithm-name publickey comment

- c. 编辑 `/etc/ssh/ssh_known_hosts` 文件，添加 `RemoteHost` 作为复制的项中的第一个字段。

```
## /etc/ssh/ssh_known_hosts File
RemoteHost <copied entry>
```

示例 17-1 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，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/ssh/shosts.equiv on machine2
machine1

## /etc/ssh/shosts.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
```

- 用户在这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
```

▼ 如何在安全 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 - 安全 Shell 端口转发必须使用 TCP 连接。安全 Shell 不支持使用 UDP 连接进行端口转发。

开始之前 您必须是 root 角色。

1 配置远程服务器的安全 Shell 设置以允许端口转发。

在 /etc/ssh/sshd_config 文件中，将 AllowTcpForwarding 的值更改为 yes。

```
# Port forwarding
AllowTcpForwarding yes
```

2 重新启动安全 Shell 服务。

```
remoteHost# svcadm restart network/ssh:default
```

有关管理持久性服务的信息，请参见《Oracle Solaris 管理：常见任务》中的第 6 章“管理服务（概述）”和 [svcadm\(1M\)](#) 手册页。

3 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
```

▼ 如何创建不同于 SSH 系统缺省值的用户和主机例外

此过程在 `/etc/ssh/sshd_config` 文件的全局部分之后添加了一个有条件的 `Match` 块。`Match` 块后的关键字-值对为指定为匹配的用户、组、主机或地址指定例外。

开始之前 您必须是 `root` 角色。

- 1 编辑 `sshd_config` 文件。
- 2 配置用户、组、主机或地址以使用不同于缺省设置的 SSH 关键字设置。
将 `Match` 块置于全局设置之后。

注- 文件的全局部分可能列出缺省设置，也可能不列出。有关缺省值，请参见 [sshd_config\(4\)](#) 手册页。

您可能有不允许使用 TCP 转发的用户。下列示例中，所有 `public` 组的用户和所有以 `test` 开头的用户名都不能使用 TCP 转发。

```
## sshd_config file
## Global settings

# Example (reflects default settings):
#
# Host *
#   ForwardAgent no
#   ForwardX11 no
#   PubkeyAuthentication yes
#   PasswordAuthentication yes
#   FallBackToRsh no
#   UseRsh no
#   BatchMode no
#   CheckHostIP yes
#   StrictHostKeyChecking ask
#   EscapeChar ~
Match Group public
    AllowTcpForwarding no
Match User test*
    AllowTcpForwarding no
```


有关 Match 块的语法信息，请参见 [sshd_config\(4\)](#) 手册页。

使用 安全 Shell (任务)

安全 Shell 可提供本地 shell 与远程 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。

使用 安全 Shell (任务列表)

以下任务列表列出了有关使用安全 Shell 的用户过程。

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安全 Shell 的访问。	第 277 页中的“如何生成用于安全 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 279 页中的“如何更改安全 Shell 私钥的口令短语”
使用安全 Shell 登录。	远程登录时，提供加密的安全 Shell 通信。此过程与使用 <code>rsh</code> 命令类似。	第 279 页中的“如何使用安全 Shell 登录到远程主机”
在不提示输入口令的情况下登录到安全 Shell。	允许使用可向安全 Shell 提供口令的代理进行登录。	第 280 页中的“如何减少安全 Shell 中的口令提示”
在安全 Shell 中使用端口转发。	指定要在基于 TCP 的安全 Shell 连接中使用的本地端口或远程端口。	第 282 页中的“如何在安全 Shell 中使用端口转发”
使用安全 Shell 复制文件。	在主机之间安全地复制文件。	第 283 页中的“如何使用安全 Shell 复制文件”
安全地连接防火墙内的主机与防火墙外的主机。	使用与 HTTP 或 SOCKS5 兼容的安全 Shell 命令连接由防火墙隔离的主机。	第 283 页中的“如何设置到防火墙外部主机的缺省连接”

▼ 如何生成用于安全 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):    <Press Return>
```

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 键入口令短语以使用密钥。

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):    <Type passphrase>
```

4 重新键入口令短语以进行确认。

```
Enter same passphrase again:    <Type passphrase>
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 279 页中的[“如何使用安全 Shell 登录到远程主机”](#)。

a. 在一行中键入以下命令 (不带反斜杠)。

```
% cat /etc/ssh/ssh_host_dsa_key.pub | ssh RemoteHost \
'cat >> ~/.ssh/known_hosts && echo "Host key copied"'
```

b. 出现提示时, 提供登录口令。

```
Enter password:    <Type password>
Host key copied
%
```

- 如果站点使用公钥进行用户验证，请在远程主机上填充 `authorized_keys` 文件。

- a. 将公钥复制到远程主机。

在一行中键入以下命令（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。

复制该文件后，将显示 "Key copied"（密钥已复制）消息。

```
Enter password:      Type login password
Key copied
myLocalHost%
```

- 7 可选少口令短语的提示。

有关过程，请参见第 280 页中的“如何减少安全 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。

▼ 如何更改安全 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 `-p` 选项的 `ssh-keygen` 命令，并回答提示。

```
myLocalHost% ssh-keygen -p
Enter file which contains the private key (/home/jdoe/.ssh/id_rsa):  <Press Return>
Enter passphrase (empty for no passphrase):  <Type passphrase>
Enter same passphrase again:  <Type passphrase>
```

其中，`-p` 用于请求更改私钥文件的口令短语。

▼ 如何使用安全 Shell 登录到远程主机

- 1 启动安全 Shell 会话。

键入 `ssh` 命令，并指定远程主机名称和您的登录信息。

```
myLocalHost% ssh myRemoteHost -l username
```

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。更新后的 `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，并继续下一步。

Are you sure you want to continue connecting(yes/no)? **yes**

3 向安全 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': *<Type passphrase>*

- ### b. 出现提示时，键入帐户口令。

```
jdoe@myRemoteHost's password: <Type password>
Last login: Wed Sep  7 09:07:49 2011 from myLocalHost
Oracle Corporation      SunOS 5.11      September 2011
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，如以下过程所述：

1 启动代理守护进程。

```
myLocalHost% eval 'ssh-agent'
Agent pid 9892
```

2 检验是否已启动代理守护进程。

```
myLocalHost% pgrep ssh-agent
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:      <Type passphrase>
Identity added: /home/jdoe/.ssh/id_rsa(/home/jdoe/.ssh/id_rsa)
myLocalHost%
```

4 启动安全 Shell 会话。

```
myLocalHost% ssh myRemoteHost -l jdoe
系统不会提示您输入口令短语。
```

示例 17-2 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` 选项用于列出该守护进程中存储的所有密钥。在会话结束时，使用 `-D` 选项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:      <Type passphrase>
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:      <Type passphrase>
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)
```

User conducts Oracle Solaris Secure Shell transactions

```
myLocalHost% ssh-add -D
Identity removed:
/home/jdoe/.ssh/id_rsa(/home/jdoe/.ssh/id_rsa.pub)
/home/jdoe/.ssh/id_dsa(DSA)
```

▼ 如何在安全 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通过安全通道实现。例如，可以指定端口 143 以通过 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，管理员必须先远程安全 Shell 服务器上启用端口转发。有关详细信息，请参见第 275 页中的“如何在安全 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。此外，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。此外，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 17-3 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，可指定 IMAP 服务器的本地端口号码，如 localhost:9143。

不要混淆 localhost 与 myLocalHost。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 17-4 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。`scp` 命令提示进行验证。有关更多信息，请参见 [scp\(1\)](#) 手册页。

也可以使用 `sftp` 安全文件传输程序。有关更多信息，请参见 [sftp\(1\)](#) 手册页。有关示例，请参见 [示例 17-5](#)。

注 – 审计服务可审计 `ft` 审计类中的 `sftp` 交易。对于 `scp`，审计服务可审计 `ssh` 会话的接入和退出。

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':    <Type passphrase>
myfile.1      25% |*****                               |    640 KB  0:20 ETA
myfile.1
```

键入口令短语后，将会显示进度指示器。请参见上述输出中的第二行。进度指示器显示以下内容：

- 文件名
- 已传输的文件百分比
- 表示已传输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) （即剩余的时间量）

示例 17-5 使用 `sftp` 命令时指定一个端口

在本示例中，用户希望 `sftp` 命令使用特定端口。用户使用 `-o` 选项来指定端口。

```
% sftp -o port=2222 guest@RemoteFileServer
```

▼ 如何设置到防火墙外部主机的缺省连接

可以使用安全 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。有关命令行选项，请参见 [示例 17-6](#)。

通常，可以通过配置文件定制 `ssh` 交互。

- 可以在 `~/.ssh/config` 中定制独立的个人文件。

- 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定制这些文件。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` 命令即可将代理命令应用到主机参数。

2 运行安全 Shell，指定外部主机。

例如，键入以下命令：

```
myLocalHost% ssh myOutsideHost
```

该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。该代理命令将替代 `ssh` 命令。

示例 17-6 通过命令行连接到防火墙外部的主机

第 283 页中的“[如何设置到防火墙外部主机的缺省连接](#)”说明了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：

- 使用 HTTP 代理命令替代 `ssh`
- 使用端口 8080 并将 `myProxyServer` 用作代理服务器
- 连接到 `myOutsideHost` 上的端口 22

安全 Shell（参考）

本章介绍了 Oracle Solaris 安全 Shell 功能中的配置选项。以下是本章中参考信息的列表。

- 第 287 页中的“典型的安全 Shell 会话”
- 第 289 页中的“安全 Shell 中的客户机和服务器配置”
- 第 290 页中的“安全 Shell 中的关键字”
- 第 295 页中的“在安全 Shell 中维护已知主机”
- 第 295 页中的“安全 Shell 文件”
- 第 297 页中的“安全 Shell 命令”

有关配置安全 Shell 的过程，请参见第 17 章，使用 [安全 Shell（任务）](#)。

典型的安全 Shell 会话

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。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。这些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验证。验证成功后，用户可以远程执行命令，并在主机之间复制数据。

安全 Shell 中的会话特征

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` 文件控制获许访问服务器的验证类型。当 sshd 守护进程启动时，服务器端行为也可以由命令行选项控制。

客户端的行为由安全 Shell 关键字按以下优先级顺序控制：

- 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 aes256-ctr,aes128-ctr,arcfour` 来覆盖优先选择 `aes128-ctr` 的系统范围配置 Ciphers。此时将首选第一种加密算法 `aes256-ctr`。

安全 Shell 中的验证和密钥交换

安全 Shell 协议支持客户机用户/主机验证和服务器主机验证。交换加密密钥以保护安全 Shell 会话。安全 Shell 提供多种验证和密钥交换方法。有些方法是可选的。在表 17-1 中列出了客户机验证机制。通过使用已知主机公钥来验证服务器。

对于验证，安全 Shell 支持用户验证和普通交互式验证，后者通常需要使用口令。安全 Shell 还支持使用用户公钥和可信主机公钥进行验证。密钥可以是 RSA 或者 DSA。会话密钥交换包括在服务器验证步骤中签名的 Diffie-Hellman 临时密钥交换。此外，安全 Shell 可以使用 GSS 凭证进行验证。

在安全 Shell 中获取 GSS 凭证

要在安全 Shell 中使用 GSS-API 进行验证，服务器必须具有 GSS-API 接受器凭证，而客户机必须具有 GSS-API 启动器凭证。提供对 `mech_dh` 和 `mech_krb5` 的支持。

对于 `mech_dh`，如果 root 运行了 `keylogin` 命令，则服务器具有 GSS-API 接受器凭证。

对于 `mech_krb5`，如果对应于服务器的主机主体在 `/etc/krb5/krb5.keytab` 中有一个有效项，则该服务器具有 GSS-API 接受器凭证。

如果执行了以下操作之一，客户机将具有 `mech_dh` 的启动器凭证：

- 运行了 `keylogin` 命令。
- 在 `pam.conf` 文件中使用了 `pam_dhkeys` 模块。

如果执行了以下操作之一，客户机将具有 `mech_krb5` 的启动器凭证：

- 运行了 `kinit` 命令。
- 在 `pam.conf` 文件中使用了 `pam_krb5` 模块。

有关如何在安全 RPC 中使用 `mech_dh` 的信息，请参见第 14 章，网络服务验证（任务）。有关使用 `mech_krb5` 的信息，请参见第 19 章，Kerberos 服务介绍。有关机制的更多信息，请参见 `mech(4)` 和 `mech_spnego(5)` 手册页。

安全 Shell 中的命令执行和数据转发

验证完成后，用户通常可以通过请求 shell 或执行命令来使用安全 Shell。通过 ssh 命令选项，用户可以发出请求。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。
在该模式下，会通过客户端终端发送或接收数据。在服务器端，会通过 shell 或命令发送数据。
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 转发都会停止。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。

安全 Shell 中的客户机和服务器配置

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。

安全 Shell 中的客户机配置

大多数情况下，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` 控制。用户配置文件 `~/.ssh/config` 可覆盖 `ssh_config` 文件中的设置。此外，用户可以通过命令行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的客户端请求。有关服务器配置设置的列表，请参见第 290 页中的“安全 Shell 中的关键字”。有关详细信息，请参见 `sshd_config(4)` 手册页。

第 290 页中的“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。有关按字母顺序排列的关键字列表及其等效的命令行覆盖项，请参见表 18-8。

安全 Shell 中的服务器配置

安全 Shell 会话的服务器端特征由 `/etc/ssh/sshd_config` 文件控制。第 290 页中的“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。

安全 Shell 中的关键字

下表列出了关键字及其缺省值（如果有）。这些关键字按字母顺序排列。应用于客户机的关键字位于 `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。已标记运行 v1 协议的安全 Shell 服务器关键字。

表 18-1 安全 Shell 配置文件中的关键字（A 到 Escape）

关键字	缺省值	位置
AllowGroups	无缺省值。	服务器
AllowTcpForwarding	yes	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
ChrootDirectory	no	服务器
Cipher	blowfish, 3des	客户机
Ciphers	aes128-ctr, aes128-cbc, 3des-cbc, blowfish-cbc, arcfour	两者
ClearAllForwardings	no	客户机
ClientAliveCountMax	3	服务器
ClientAliveInterval	0	服务器
Compression	no	两者
CompressionLevel	无缺省值。	客户机
ConnectionAttempts	1	客户机
ConnectTimeout	系统 TCP 超时	客户机
DenyGroups	无缺省值	服务器
DenyUsers	无缺省值	服务器
DisableBanner	no	客户机
DynamicForward	无缺省值。	客户机

表 18-1 安全 Shell 配置文件中的关键字（A 到 Escape）（续）

关键字	缺省值	位置
EscapeChar	~	客户机

表 18-2 安全 Shell 配置文件中的关键字（Fall 至 Local）

关键字	缺省值	位置
FallBackToRsh	no	客户机
ForwardAgent	no	客户机
ForwardX11	no	客户机
ForwardX11	yes	客户机
GatewayPorts	no	两者
GlobalKnownHostsFile	/etc/ssh/ssh_known_hosts	客户机
GSSAPIAuthentication	yes	两者
GSSAPIDelegateCredentials	no	客户机
GSSAPIKeyExchange	yes	两者
GSSAPIStoreDelegateCredentials	yes	服务器
HashKnownHosts	no	客户机
Host	*有关更多信息，请参见第 294 页中的“安全 Shell 中主机特定的参数”。	客户机
HostbasedAuthentication	no	两者
HostbasedUsesNameFromPacketOnly	no	服务器
HostKey	/etc/ssh/ssh_host_key	服务器，v1
HostKey	/etc/ssh/host_rsa_key, /etc/ssh/host_dsa_key	服务器
HostKeyAlgorithms	ssh-rsa, ssh-dss	客户机
HostKeyAlias	无缺省值。	客户机
HostName	无缺省值。	客户机
IdentityFile	~/.ssh/id_dsa, ~/.ssh/id_rsa	客户机
IgnoreIfUnknown	无缺省值	客户机
IgnoreRhosts	yes	服务器

表 18-2 安全 Shell 配置文件中的关键字（Fall 至 Local）（续）

关键字	缺省值	位置
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 18-3 安全 Shell 配置文件中的关键字（Login 至 R）

关键字	缺省值	位置
LoginGraceTime	120（秒）	服务器
LogLevel	info	两者
LookupClientHostnames	yes	服务器
MACs	hmac-sha1,hmac-md5	两者
Match	无缺省值	服务器
MaxStartups	10:30:60	服务器
NoHostAuthenticationForLocalHost	no	客户机
NumberOfPasswordPrompts	3	客户机
PAMServiceName	无缺省值	服务器
PAMServicePrefix	无缺省值	服务器
PasswordAuthentication	yes	两者
PermitEmptyPasswords	no	服务器
PermitRootLogin	no	服务器
PermitUserEnvironment	no	服务器
PidFile	/system/volatile/sshd.pid	服务器
Port	22	两者
PreferredAuthentications	hostbased,publickey,keyboard-interactive,password	客户机
PreUserauthHook	无缺省值	服务器

表 18-3 安全 Shell 配置文件中的关键字（Login 至 R）（续）

关键字	缺省值	位置
PrintLastLog	yes	服务器
PrintMotd	no	服务器
Protocol	2,1	两者
ProxyCommand	无缺省值。	客户机
PubkeyAuthentication	yes	两者
RekeyLimit	1G 至 4G	客户机
RemoteForward	无缺省值。	客户机
RhostsAuthentication	no	服务器， v1
RhostsRSAAuthentication	no	服务器， v1
RSAAuthentication	no	服务器， v1

表 18-4 安全 Shell 配置文件中的关键字（S 至 X）

关键字	缺省值	位置
ServerAliveCountMax	3	客户机
ServerAliveInterval	0	客户机
ServerKeyBits	512 至 768	服务器， v1
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
UseOpenSSLEngine	yes	两者
UsePrivilegedPort	no	两者
User	无缺省值	客户机
UserKnownHostsFile	~/.ssh/known_hosts	客户机
UseRsh	no	客户机
VerifyReverseMapping	no	服务器
X11DisplayOffset	10	服务器
X11Forwarding	yes	服务器

表 18-4 安全 Shell 配置文件中的关键字（S 至 X）（续）

关键字	缺省值	位置
X11UseLocalHost	yes	服务器
XAuthLocation	/usr/openwin/bin/xauth	两者

安全 Shell 中主机特定的参数

如果适合于使不同的本地主机具有不同的安全 Shell 特征，则管理员可以在 `/etc/ssh/ssh_config` 文件中定义不同的参数集，从而根据主机或正则表达式进行应用。此任务是通过按 `Host` 关键字对该文件中的项进行分组完成的。如果未使用 `Host` 关键字，则客户机配置文件中的项将应用于用户正在使用的任意本地主机。

安全 Shell 和登录环境变量

如果在 `sshd_config` 文件中未设置下列安全 Shell 关键字，它们将从 `/etc/default/login` 文件中的等效项获取其值。

<code>/etc/default/login</code> 中的项	<code>sshd_config</code> 中的关键字和值
<code>CONSOLE=*</code>	<code>PermitRootLogin=without-password</code>
<code>#CONSOLE=*</code>	<code>PermitRootLogin=yes</code>
<code>PASSREQ=YES</code>	<code>PermitEmptyPasswords=no</code>
<code>PASSREQ=NO</code>	<code>PermitEmptyPasswords=yes</code>
<code>#PASSREQ</code>	<code>PermitEmptyPasswords=no</code>
<code>TIMEOUT=secs</code>	<code>LoginGraceTime=secs</code>
<code>#TIMEOUT</code>	<code>LoginGraceTime=120</code>
<code>RETRIES</code> 和 <code>SYSLOG_FAILED_LOGINS</code>	仅适用于 <code>password</code> 和 <code>keyboard-interactive</code> 验证方法。

当用户的登录 shell 中的初始化脚本设置了下列变量时，`sshd` 守护进程会使用这些值。如果未设置这些变量，守护进程将使用缺省值。

TIMEZONE	控制 TZ 环境变量的设置。如果未设置这些变量， <code>sshd</code> 守护进程在启动时会使用 TZ 值。
ALTSHELL	控制 SHELL 环境变量的设置。缺省值为 <code>ALTSHELL=YES</code> ，其中 <code>sshd</code> 守护进程使用用户的 shell 值。当 <code>ALTSHELL=NO</code> 时，表明 SHELL 值未设置。
PATH	控制 PATH 环境变量的设置。如果未设置该值，缺省路径为 <code>/usr/bin</code> 。

SUPATH 控制 root 的 PATH 环境变量的设置。如果未设置该值，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。

在安全 Shell 中维护已知主机

对于需要与其他主机安全通信的每个主机，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。虽然可使用脚本来更新 /etc/ssh/ssh_known_hosts 文件，但强烈建议不要使用这种做法，因为脚本会导致严重安全漏洞。

/etc/ssh/ssh_known_hosts 文件只能通过安全机制分发，如下所述：

- 通过安全的连接，如安全 Shell、IPsec 或者来自已知可信计算机的基于 Kerberos 的 ftp
- 安装系统时

为了避免入侵者通过在 known_hosts 文件中插入伪造的公钥来获得访问权限的可能性，您应使用 ssh_known_hosts 文件的已知可信任源。可以在安装期间分发 ssh_known_hosts 文件。然后，使用 scp 命令的脚本可用于引入最新的版本。

安全 Shell 文件

下表列出了重要的安全 Shell 文件和建议的文件权限。

表 18-5 安全 Shell 文件

文件名	说明	建议的权限和所有者
/etc/ssh/sshd_config	包含安全 Shell 守护进程 sshd 的配置数据。	-rw-r--r-- root
/etc/ssh/ssh_host_dsa_key 或 /etc/ssh/ssh_host_rsa_key	包含主机私钥。	-rw----- root
host-private-key.pub	包含主机公钥，例如 /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
/system/volatile/sshd.pid	包含安全 Shell 守护进程 sshd 的进程 ID。如果多个守护进程正在运行，该文件包含最后启动的守护进程。	-rw-r--r-- root
~/.ssh/authorized_keys	包含获许以用户帐户登录的用户的公钥。	-rw-r--r-- username
/etc/ssh/ssh_known_hosts	包含客户机可以安全与之通信的所有主机的主机公钥。该文件由管理员填充。	-rw-r--r-- root

表 18-5 安全 Shell 文件 (续)

文件名	说明	建议的权限和所有者
~/.ssh/known_hosts	包含客户机可以安全与之通信的所有主机的主机公钥。该文件由系统自动维护。当用户与未知主机连接时，会将该远程主机密钥添加到文件中。	-rw-r--r-- username
/etc/default/login	如果未设置对应的 sshd_config 参数，则为 sshd 守护进程提供缺省值。	-r--r--r-- root
/etc/nologin	如果该文件存在，sshd 守护进程仅允许 root 登录。系统会向尝试登录的用户显示此文件的内容。	-rw-r--r-- root
~/.rhosts	包含指定用户不使用口令也可登录的主机的主机/用户名对。该文件还用于 rlogind 和 rshd 守护进程。	-rw-r--r-- username
~/.shosts	包含指定用户不使用口令也可登录的主机的主机/用户名对。该文件不用于其他实用程序。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。该文件还用于 rlogind 和 rshd 守护进程。	-rw-r--r-- root
/etc/ssh/shosts.equiv	包含基于主机的验证中使用的主机。该文件不用于其他实用程序。	-rw-r--r-- root
~/.ssh/environment	包含登录时的初始指定。缺省情况下，不读取该文件。要读取此文件，必须将 sshd_config 文件中的 PermitUserEnvironment 关键字设为 yes。	-rw-r--r-- username
~/.ssh/rc	包含在用户 shell 启动之前运行的初始化例程。有关初始化例程样例，请参见 sshd(1M) 手册页。	-rw-r--r-- username
/etc/ssh/sshrd	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，该设置将覆盖系统设置。	-rw-r--r-- username

下表列出了可被关键字或命令选项覆盖的安全 Shell 文件。

表 18-6 覆盖安全 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
/etc/ssh/host_rsa_key	HostKey	
/etc/ssh/host_dsa_key		

表 18-6 覆盖安全 Shell 文件的位置 (续)

文件名	关键字覆盖	命令行覆盖
~/.ssh/identity	IdentityFile	ssh -i <i>id-file</i>
~/.ssh/id_dsa,~/.ssh/id_rsa		scp -i <i>id-file</i>
~/.ssh/authorized_keys	AuthorizedKeysFile	
/etc/ssh/ssh_known_hosts	GlobalKnownHostsFile	
~/.ssh/known_hosts	UserKnownHostsFile	
	IgnoreUserKnownHosts	

安全 Shell 命令

下表概括介绍了主要的安全 Shell 命令。

表 18-7 安全 Shell 中的命令

命令的手册页	说明
ssh(1)	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。该命令是 rlogin 和 rsh 命令的安全 Shell 替代项。ssh 命令实现在两个不可信主机之间通过不安全网络进行安全的加密通信。X11 连接和任意的 TCP/IP 端口也可以通过安全通道转发。
sshd(1M)	安全 Shell 的守护进程。此守护进程侦听客户机连接，并允许在两个不可信主机之间通过不安全网络进行安全加密通信。
ssh-add(1)	将 RSA 或 DSA 标识添加到验证代理 ssh-agent。标识也称为 密钥 。
ssh-agent(1)	保存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都作为 ssh-agent 程序的客户机启动。当用户使用 ssh 命令登录其他系统时，通过使用环境变量，可以定位代理并将其用于验证。
ssh-keygen(1)	生成并管理安全 Shell 的验证密钥。
ssh-keyscan(1)	收集多个安全 Shell 主机的公钥。帮助生成和检验 ssh_known_hosts 文件。
ssh-keysign(1M)	由 ssh 命令用于访问本地主机中的主机密钥。生成在使用安全 Shell v2 进行基于主机的验证时所需的数字签名。此命令由 ssh 命令而非用户调用。
scp(1)	通过加密的 ssh 传输在网络上的主机之间安全复制文件。与 rcp 命令不同，如果验证要求提供口令信息，则 scp 命令会提示输入口令或口令短语。
sftp(1)	类似于 ftp 命令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。该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。

下表列出了覆盖安全 Shell 关键字的命令选项。关键字在 ssh_config 和 sshd_config 文件中指定。

表 18-8 安全 Shell 关键字的命令行等效项

关键字	ssh 命令行覆盖	scp 命令行覆盖
BatchMode		scp -B
BindAddress	ssh -b <i>bind-addr</i>	scp -a <i>bind-addr</i>
Cipher	ssh -c <i>cipher</i>	scp -c <i>cipher</i>
Ciphers	ssh -c <i>cipher-spec</i>	scp -c <i>cipher-spec</i>
Compression	ssh -C	scp -C
DynamicForward	ssh -D <i>SOCKS4-port</i>	
EscapeChar	ssh -e <i>escape-char</i>	
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 <i>localport:remotehost:remoteport</i>	
MACS	ssh -m <i>mac-spec</i>	
Port	ssh -p <i>port</i>	scp -P <i>port</i>
Protocol	ssh -2 (仅用于 v2)	
RemoteForward	ssh -R <i>remoteport:localhost:localport</i>	

第 6 部分

Kerberos 服务

本部分在以下各章中提供有关配置、管理和使用 Kerberos 服务的信息：

- 第 19 章，Kerberos 服务介绍
- 第 20 章，规划 Kerberos 服务
- 第 21 章，配置 Kerberos 服务（任务）
- 第 22 章，Kerberos 错误消息和故障排除
- 第 23 章，管理 Kerberos 主体和策略（任务）
- 第 24 章，使用 Kerberos 应用程序（任务）
- 第 25 章，Kerberos 服务（参考）

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 301 页中的“什么是 Kerberos 服务？”
- 第 302 页中的“Kerberos 服务的工作原理”
- 第 308 页中的“Kerberos 安全服务”
- 第 308 页中的“各种 Kerberos 发行版的组件”

什么是 Kerberos 服务？

“Kerberos 服务”是一种客户机/服务器体系结构，用于在网络上提供安全事务。该服务可提供功能强大的用户验证以及完整性和保密性。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，这样，管理员便可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要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，便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）。

Oracle Solaris 发行版的 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 Oracle 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 Oracle 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 Oracle Solaris 发行版的 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 Oracle 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，即使当前应用程序运行的系统上未启用 Kerberos 服务，这些应用程序仍能工作。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适合其需求的任何安全机制。

Kerberos 服务的工作原理

下面概述了 Kerberos 验证系统。有关更详细的说明，请参见第 468 页中的[“Kerberos 验证系统的工作原理”](#)。

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特权。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** (Key Distribution Center, 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。

票证具有与之关联的特定属性。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 469 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。

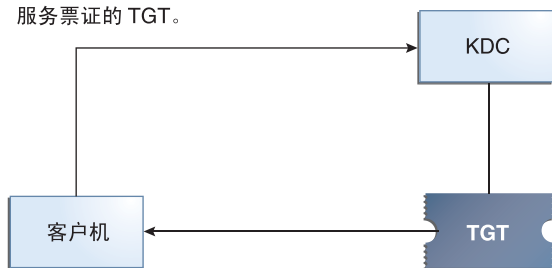
初始验证：票证授予票证

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 19-1 Kerberos 会话的初始验证

1. 登录（或使用 `kinit`）时，客户机请求允许其获取服务票证的 TGT。



3. 客户机使用口令解密 TGT，进而提供标识；现在可以使用 TGT 获取其他票证。

2. KDC 检查数据库，发送 TGT。

TGT = 票证授予票证
KDC = 密钥分发中心

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。只要入场券未过期，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。

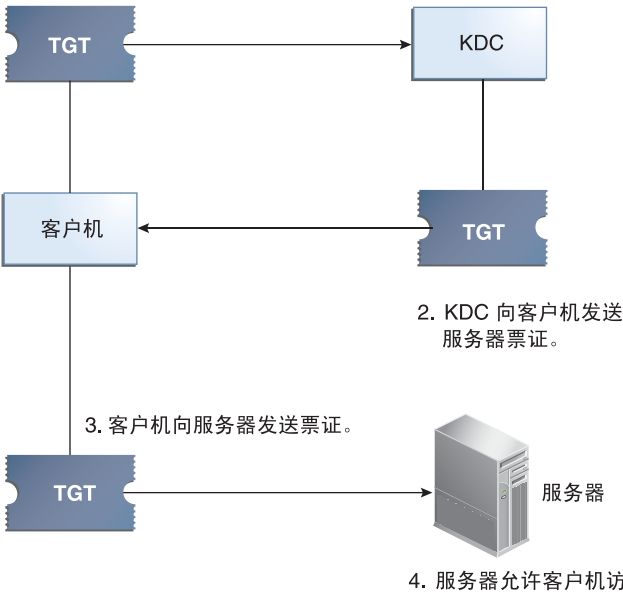
3. 拥有有效的票证授予票证后，只要该票证授予票证未过期，客户机便可以请求所有类型的网络操作（如 `rlogin` 或 `telnet`）的票证。此票证的有效期限通常为几个小时。每次客户机执行唯一的网络操作时，都将从 KDC 请求该操作的票证。

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。

图 19-2 使用 Kerberos 验证获取对服务的访问权

1. 客户机请求服务器票证；
向 KDC 发送 TGT 作为
标识的证明。



TGT = 票证授予票证
KDC = 密钥分发中心

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，假设用户 `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，假设用户 `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`
- `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 305 页中的“Kerberos 主体”。

第 455 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 可以为其分配票证的唯一标识。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`。在此示例中：

- `joe` 是主名称。主名称可以是用户名（如此处所示）或服务（如 `nfs`）。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` 和 `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` 将其自身与其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 306 页中的“Kerberos 领域”中介绍。

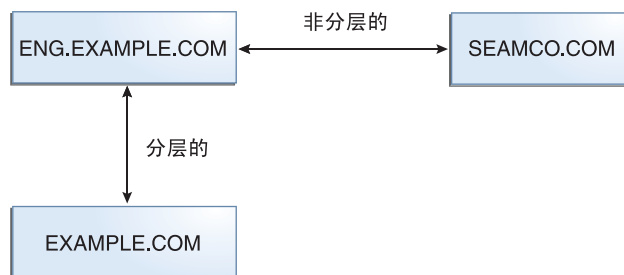
以下都是有效的主体名称：

- `joe`
- `joe/admin`
- `joe/admin@ENG.EXAMPLE.COM`
- `nfs/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 19-3 显示了各领域相互之间的关系。有些领域是层次化的，其中，一个领域是另一个领域的超集。另外一些领域是非层次化（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 19-3 Kerberos 领域



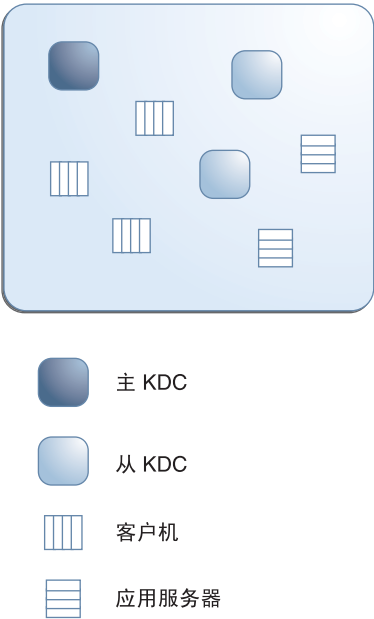
Kerberos 服务器

每个领域都必须包括一台用于维护主体数据库主副本的服务器。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 Kerberos **应用服务器**。该服务器提供对基于 Kerberos 的服务（如 ftp、telnet、rsh 和 NFS）的访问。如果安装了 SEAM 1.0 或 1.0.1，则领域可能会包括 Kerberos 网络应用服务器，但此软件不随这些发行版一起提供。

下图显示了一个假设的领域可能包含的内容。

图 19-4 典型的 Kerberos 领域
EXAMPLE.COM



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。

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。随着 Oracle Solaris 软件中包括的 SEAM 产品组件

越来越多，SEAM 发行版的内容会越来越少。从 Oracle Solaris 10 发行版开始，包括了 SEAM 产品的所有组件，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 19-1 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
从 Oracle Solaris 10 发行版开始的 Kerberos 服务	带有增强功能的 Kerberos 服务的完整发行版

有关 Oracle Solaris 10 发行版中包括的增强功能的更多信息，请参见[Kerberos 组件](#)。

Kerberos 组件

与 MIT 发布的 Kerberos V5 产品类似，Oracle Solaris 发行版的 Kerberos 服务也包括以下内容：

- 密钥分发中心 (Key Distribution Center, KDC) :
 - Kerberos 数据库管理守护进程—`kadmind`。
 - Kerberos 票证处理守护进程—`krb5kdc`。
 - 数据库管理程序—`kadmin`（仅主 KDC）、`kadmin.local` 和 `kdb5_util`。
 - 数据库传播软件—`kprop`（仅从 KDC）和 `kpropd`。
- 用于管理凭证的用户程序—`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`、`rlogin`、`rsh`、`ssh` 和 `telnet`。
- 远程应用程序守护进程—`ftpd`、`rlogind`、`rshd`、`sshd` 和 `telnetd`。
- 密钥表管理实用程序—`ktutil`。

- 通用安全服务应用编程接口 (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。GSS-API 使用标准接口，允许将应用程序移植到多种操作系统上。使用 GSS-API，应用程序可包括完整性和保密性安全服务以及验证。ftp 和 ssh 均使用 GSS-API。
- RPCSEC_GSS 应用编程接口 (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。

此外，Oracle Solaris 发行版的 Kerberos 服务还包括以下内容：

- 一个基于 GUI 的 Kerberos 管理工具 (gkadmin) — 允许您管理主体和主体策略。这一基于 Java 技术的 GUI 是 kadmin 命令的替代方法。
- PAM 的 Kerberos V5 服务模块 — 为 Kerberos 服务提供验证、帐户管理、会话管理和口令管理。使用该模块可使 Kerberos 验证对用户而言透明化。
- 内核模块 — 提供 kerberos 服务的基于内核的实现以供 NFS 服务使用，这可大大提高性能。

关于 Oracle Solaris 11 发行版中的 Kerberos

本节列出了 Oracle Solaris 11 发行版中的可用更改。

- Kerberos 软件已与 MIT 1.8 发行版进行了同步。包括以下功能：
 - 缺省情况下，arcfour-hmac-md5-exp、des-cbc-md5 和 des-cbc-crc 弱加密类型处于禁用状态。可以在 /etc/krb5/krb5.conf 文件中添加 allow_weak_crypto = true 声明以便允许使用更弱的加密算法。
 - 在 /etc/krb5/krb5.conf 文件中，permitted_enctypes 关系可以使用可选的 DEFAULT 关键字以及 + 或 - enctype_family 来从缺省设置中添加或删除指定的加密类型。
 - 在大多数情况下，都可以消除客户端对 domain_realm 映射表的需要，方法是在 KDC 中实现最小引用支持，并通过该协议向客户机提供映射信息。客户机可以在没有 domain_realm 映射表的情况下运行，方法是将对服务主体 name service/canonical-fqdn@LOCAL.REALM 的请求发送到本地 KDC 并请求引用。此功能可能会受限于特定名称类型或特定格式的服务主体名称。KDC 只能使用其 domain_realm 映射表。无法引入对 DNS 的分块查询。
 - 如果是将 LDAP 后端用于 Kerberos 数据库，则可以为主体条目创建别名。如果服务可供不同的主机名访问，或者 DNS 不可用于标准化主机名（意味着使用的是简短形式），则主体别名支持会很有用。可以对已知的各种服务主体名称使用一个别名，对于系统密钥表文件中的实际服务主体，系统只需一组密钥即可。
 - 可以使用 kvno 实用程序诊断存储在 /etc/krb5/krb5.keytab 中的服务主体密钥存在的问题。

- `kadmin ktadd` 命令支持 `-norandkey` 选项，后者阻止 `kadmin` 命令创建新的随机密钥。如果您要为一个包含口令派生密钥的主体创建密钥表，`-norandkey` 选项可能会很有用。可以创建一个密钥表用于运行 `kinit` 命令，而无需指定口令。
- 在给定时间限制内，预验证失败一定次数后，可以锁定主体。有关更多信息，请参见第 366 页中的“如何配置帐户锁定”。
- `OK_AS_DELEGATE` 标志使 KDC 能够将本地领域策略传递给客户机，该策略与是否信任中间服务器接受委托凭证有关。有关更多信息，请参见第 319 页中的“委托服务的信任”。
- 为 Kerberos 添加了一组用户级静态定义的跟踪点。这些探测器提供对 Kerberos 协议消息的逻辑视图。有关示例，请参见第 409 页中的“对 Kerberos 服务使用 DTrace”。
- `kclient` 脚本已得到增强。脚本包括加入 Microsoft Active Directory 服务器的功能。有关说明，请参见第 354 页中的“如何交互配置 Kerberos 客户机”和第 357 页中的“如何为 Active Directory 服务器配置 Kerberos 客户机”。此外，脚本还包括 `-T` 选项，可用于识别客户机的 KDC 服务器类型。该脚本的所有选项在 `kclient(1M)` 手册页中均有介绍。
- 不再需要 `/etc/krb5/kadm5.keytab` 文件。存储在该文件中的密钥现在可以直接从 Kerberos 数据库中读取。
- 添加了对使用 LDAP 从目录服务器访问 Kerberos 主体和策略记录的支持。此更改简化了管理，并可提供更高的可用性，具体取决于 KDC 和目录服务器的部署。有关与 LDAP 相关的过程列表，请参见第 387 页中的“在 LDAP 目录服务器上管理 KDC”。
- 新增的 `kdcmgr` 命令可用于自动或交互设置任何 KDC。该命令创建主 KDC 服务器和从 KDC 服务器。此外，与 `status` 选项结合使用时，`kdcmgr` 命令显示任何安装在本地主机上的 KDC 的相关信息。有关自动和交互式过程的链接，请参见表 21-1。
- 此发行版中添加了对无需额外设置的 Oracle Solaris 客户机的支持。对 Kerberos 服务和一些缺省值进行了更改。Kerberos 客户机可在进行了适当配置的环境中采用无客户端配置。有关更多信息，请参见第 318 页中的“客户机配置选项”。

规划 Kerberos 服务

参与安装和维护 Kerberos 服务的管理员应认真学习本章。本章介绍管理员在安装或配置该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列表：

- 第 313 页中的“为什么要规划 Kerberos 部署？”
- 第 314 页中的“规划 Kerberos 领域”
- 第 315 页中的“将主机名映射到领域”
- 第 315 页中的“客户机与服务主体名称”
- 第 316 页中的“KDC 端口和管理服务端口”
- 第 316 页中的“从 KDC 数量”
- 第 317 页中的“要使用的数据库传播系统”
- 第 318 页中的“领域内的时钟同步”
- 第 318 页中的“客户机配置选项”
- 第 319 页中的“提高客户机登录安全性”
- 第 319 页中的“KDC 配置选项”
- 第 319 页中的“委托服务的信任”
- 第 320 页中的“Kerberos 加密类型”
- 第 320 页中的“图形化 Kerberos 管理工具中的联机帮助 URL”

为什么要规划 Kerberos 部署？

安装 Kerberos 服务之前，必须先解决几个配置问题。尽管在初始安装后可以更改配置，但有些更改执行起来非常困难。此外，有些更改需要重建 KDC，所以在规划 Kerberos 配置时最好考虑长期目标。

部署 Kerberos 基础结构涉及到安装 KDC、为主机创建密钥和迁移用户等任务。重新配置 Kerberos 部署会像执行初始部署一样难，因此，请您仔细规划部署，以避免将来不得不重新配置。

规划 Kerberos 领域

领域是类似于域的逻辑网络，用于定义相同主 KDC 下的一组系统。与建立 DNS 域名一样，在配置 Kerberos 服务之前，应解决领域名称、领域数和每个领域的大小，以及各领域之间的关系（为了能够进行跨领域验证）等相关问题。

领域名称

领域名称可以包括任意 ASCII 字符串。通常，领域名称与您的 DNS 域名相同，只是领域名称采用大写。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，或是选择使用不同的字符串，则可以使用任意字符串。但是，配置过程需要更多的工作。采用符合标准 Internet 命名结构的领域名称是明智之举。

领域数量

安装所需的领域数量由以下因素决定：

- 支持的客户机数量。如果一个领域内客户机过多，管理起来会非常困难，最终仍然需要您分割领域。确定可以支持的客户机数量的主要因素如下：
 - 每台客户机产生的 Kerberos 流量
 - 物理网络带宽
 - 主机速度

因为每种安装都有不同的限制，所以没有任何规则可以确定客户机的最大数量。

- 客户机间的距离。如果客户机分别位于不同的地理区域，设置若干个小领域可能更有效果。
- 可作为 KDC 安装的主机数量。每个领域至少有两台 KDC 服务器，一台主服务器，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捆绑在一起。可以在这些领域之间建立层次化关系，以便提供指向相关域的自动路径。当然，层次化链中的所有领域都必须配置正确。自动路径可以减轻管理负担。但是，如果域有许多层，您可能不希望使用缺省路径，因为这涉及到太多的事务。

您也可以选择直接建立信任关系。当两个层次化领域之间存在的层太多或不存在层次化关系时，最好使用直接信任关系。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，这需要一些额外的工作。直接信任关系又称为可传递关系。有关介绍，请参见第 306 页中的“Kerberos 领域”。有关针对多个领域的配置步骤，请参见第 341 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单个主机定义这些映射。

也可以使用 DNS 来查找有关 KDC 的信息。使用 DNS 可以更轻松地更改信息，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。

Solaris Kerberos 客户机可以更好地与 Active Directory 服务器交互操作。可以将 Active Directory 服务器配置为提供领域到主机的映射。

客户机与服务主体名称

使用 Kerberos 服务时，所有主机上都必须启用 DNS。使用 DNS 时，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`，那么主机的主体名称应该是 `host/boston.example.com@EXAMPLE.COM`。本书中的示例要求每台主机都已配置 DNS，且使用 FQDN。

在为关联服务构建服务主体时，Kerberos 服务将通过 DNS 标准化主机别名，并使用标准化的格式 (cname)。因此，创建服务主体时，服务主体名称的主机名部分应该是托管服务的系统的主机名标准格式。

以下是 Kerberos 服务如何标准化主机名的示例：如果用户运行了命令 `"ssh alpha.example.com"`，其中 `alpha.example.com` 是标准化名称 `beta.example.com` 的 DNS 主机别名。当 `ssh` 调用 Kerberos 并请求 `alpha.example.com` 的主机服务票证时，Kerberos 服务会将 `alpha.example.com` 标准化为 `beta.example.com`，并从 KDC 请求服务主体 `"host/beta.example.com"` 的票证。

对于包括主机 FQDN 的主体名称，与 `/etc/resolv.conf` 文件中描述 DNS 域名的字符串相匹配是非常重要的。为主体指定 FQDN 时，Kerberos 服务要求 DNS 域名使用小写字母。DNS 域名可以包括大写和小写字母，但是创建主机主体时只能使用小写字母。例如，DNS 域名 `example.com`、`Example.COM` 或其他任何变体都是等效的。主机的主体名称仍将是 `host/boston.example.com@EXAMPLE.COM`。

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。`kdb5_util`、`kadmind` 和 `kpropd` 守护进程，以及 `kprop` 命令都配置为依赖于 DNS 服务。要充分利用 Kerberos 服务和 SMF 提供的功能，必须在所有主机上启用 DNS 客户机服务。

KDC 端口和管理服务端口

缺省情况下，端口 88 和端口 750 供 KDC 使用，端口 749 供 KDC 管理守护进程使用。可以使用不同的端口号。但是，如果更改端口号，则必须更改每台客户机上的 `/etc/services` 和 `/etc/krb5/krb5.conf` 文件。除这些文件外，还必须更新每个 KDC 上的 `/etc/krb5/kdc.conf` 文件。

从 KDC 数量

与主 KDC 一样，从 KDC 也能为客户机生成凭证。如果主 KDC 不可用，从 KDC 将会提供备份。每个领域至少应有一个从 KDC。可能还需要其他的从 KDC，视以下因素而定：

- 领域中的物理段数。通常，应至少将网络设置为领域中其他组件不可用情况下，每个段仍可工作。为此，必须能够从每个段访问 KDC。该实例中的 KDC 可以是主 KDC 或从 KDC。
- 领域中的客户机数量。添加更多的从 KDC 服务器可以减少当前服务器的负荷。

添加的从 KDC 有可能过量。请记住，KDC 数据库一定会传播到每台服务器，因此，安装的 KDC 服务器越多，更新整个领域中的数据所用的时间就越长。另外，因为每个从 KDC 都会保存 KDC 数据库的副本，所以增加从 KDC 会增大安全违规的风险。

此外，可以轻松地将一个或多个从 KDC 配置为与主 KDC 交换。以这种方式配置至少一个从 KDC 的优点是，如果主 KDC 由于任何原因而出故障，可以使用一个可轻松交换为主 KDC 的预配置系统。关于如何配置可交换从 KDC 的说明，请参见第 368 页中的“交换主 KDC 服务器与从 KDC 服务器”。

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。将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 348 页中的“如何创建凭证表”](#)
- [第 348 页中的“如何向凭证表中添加单个项”](#)
- [第 349 页中的“如何提供各领域之间的凭证映射”](#)
- [第 408 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。新的主体口令与 UNIX 口令相同。请参见[第 364 页中的“如何在 Kerberos 领域中配置用户自动迁移”](#)，了解有关如何使用 `pam_krb5_migrate` 模块的说明。

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。可以将数据库传播配置为以增量方式进行。增量过程仅将更新的信息（而不是整个数据库）传播到从 KDC。有关数据库传播的更多信息，请参见[第 372 页中的“管理 Kerberos 数据库”](#)。

如果不使用增量传播，要解决的首要问题之一是，多久更新一次从 KDC。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。

在一个领域包含多个 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。该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。有关该策略的完整说明，请参见[第 383 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部同步其内部时钟。该功能称为**时钟相位差**，它提供另一种 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 (NTP) 软件。有关更多信息，请参见[第 367 页中的“同步 KDC 与 Kerberos 客户机的时钟”](#)。也可以采用其他方式同步时钟，并非一定要使用 NTP。但是，由于存在时钟相位差，应使用某种形式的同步来防止访问失败。

客户机配置选项

kcclient 配置实用程序是 Solaris 10 发行版的新功能。该实用程序既可以在交互模式下运行，也可以在非交互模式下运行。在交互模式下，系统会提示用户输入 Kerberos 特定的参数值，以使用户在配置客户机时能够更改现有的安装。在非交互模式下，将会使用包含事先设置的参数值的文件。另外，在非交互模式下可以使用命令行选项。无论是交互模式还是非交互模式，需要的步骤都比手动过程要少，因此可以加速操作过程，也更不容易出错。

Solaris Express Developer Edition 1/08 发行版做了一些更改，以配合零配置 Kerberos 客户机。如果您的环境遵照了下列规则，将无需对 Solaris Kerberos 客户机执行显式配置步骤：

- DNS 配置为返回 KDC 的 SRV 记录。
- 领域名称与 DNS 域名匹配，或者 KDC 支持引用。
- Kerberos 客户机不需要 keytab 文件。

有些情况下，最好显式配置 Kerberos 客户机。

- 如果不使用引用，零配置逻辑会根据主机的 DNS 域名来确定领域。这会带来一定的安全风险，但风险比启用 `dns_lookup_realm` 要小得多。
- `pam_krb5` 模块依赖于 `keytab` 中的主机密钥项。这一要求可以在 `krb5.conf` 文件中禁用，但是出于安全原因，不建议这样做。请参见 [krb5.conf\(4\)](#) 手册页。
- 零配置过程比直接配置效率要低，且对 DNS 有更强的依赖性。该过程比直接配置客户机执行更多的 DNS 查找。

有关所有客户机配置过程的说明，请参见[第 351 页中的“配置 Kerberos 客户机”](#)。

提高客户机登录安全性

登录客户机后，系统会使用 `pam_krb5` 模块验证发出最新 TGT 的 KDC 是否为发出 `/etc/krb5/krb5.keytab` 中存储的客户机主机主体的 KDC。在验证栈中配置 `pam_krb5` 模块后，该模块即可验证 KDC。对于有些配置，例如不存储客户机主机主体的 DHCP 客户机，则需要禁用该项检查。要关闭此检查，必须将 `krb5.conf` 文件中的 `verify_ap_req_nofail` 选项设置为 `false`。有关更多信息，请参见第 362 页中的“如何禁用票务授予票证的验证”。

KDC 配置选项

有几种方法可用于配置 KDC。最简单的方法是使用 `kdcmgmt` 实用程序以自动或交互方式配置 KDC。自动方式要求使用命令行选项来定义配置参数。该方法尤其适用于脚本。交互方式会提示您提供所有所需信息。有关指向使用该命令的说明的链接，请参见表 21-1。

还提供了对使用 LDAP 来管理 Kerberos 的数据库文件的支持。有关说明，请参见第 329 页中的“如何配置 KDC 以使用 LDAP 数据服务器”。使用 LDAP 可以简化那些需要更好协调 Kerberos 数据库及其现有目录服务器设置的站点的管理工作。

委托服务的信任

对于某些应用程序，客户机可能需要将授权委托给代其联系其他服务的某个服务器。客户机必须将凭证转发给中间服务器。客户机获取服务器服务票证的功能不会向客户机传递任何有关是否应信任服务器接受委托凭证的信息。`kadmin` 命令的 `ok_to_auth_as_delegate` 选项为 KDC 将本地领域策略传递给客户机提供了一种方法，该策略与是否应信任中间服务器接受此凭证有关。

KDC 回复的加密部分中凭证票证标志的副本可能会设置 `ok_to_auth_as_delegate` 选项，以向客户机表明领域策略已确定票证中指定的服务器是合适的委托接受者。客户机可以使用该信息的存在来确定是否要将凭证委托（通过授予代理或转发 TGT）给该服务器。设置此选项时，管理员必须考虑运行服务的服务器的安全性和位置，以及服务是否要求使用委托凭证。

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所用的加密算法和模式。以下是支持的加密类型：

- des-cbc-md5
- des-cbc-crc
- des3-cbc-sha1-kd
- arcfour-hmac-md5
- arcfour-hmac-md5-exp
- aes128-cts-hmac-sha1-96
- aes256-cts-hmac-sha1-96

注 – 在 Solaris 10 8/07 之前的发行版中，如果安装了非捆绑的强加密软件包，则可以将 aes256-cts-hmac-sha1-96 加密类型用于 Kerberos 服务。

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）和客户机之间存在交互，因此很难在现有数据库上更改加密类型。除非要重新创建数据库，否则请不要设置这些参数。有关更多信息，请参阅第 472 页中的“使用 Kerberos 加密类型”。

注 – 如果安装的主 KDC 未运行 Solaris 10 发行版，在升级主 KDC 之前，必须将从 KDC 升级到 Solaris 10 发行版。Solaris 10 主 KDC 使用新的加密类型，而早期版本的从 KDC 将无法处理这些加密类型。

在 Oracle Solaris 11 发行版中，缺省情况下，arcfour-hmac-md5-exp、des-cbc-md5 和 des-cbc-crc 弱加密类型处于禁用状态。如果您需要继续使用这些加密类型，则应在 /etc/krb5/krb5.conf 文件的 libdefaults 部分中设置 allow_weak_crypto = true。

图形化 Kerberos 管理工具中的联机帮助 URL

图形化 Kerberos 管理工具 gkadmin 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。可以在任何适当的服务器上安装本手册的 HTML 版本。您也可以根据需要使用 <http://www.oracle.com/technetwork/indexes/documentation/index.html> 中的文档集。

将主机配置为使用 Kerberos 服务时，需要在 krb5.conf 文件中指定 URL。该 URL 应指向本书的“管理 Kerberos 主体和策略（任务）”一章中标题为第 412 页中的“SEAM Tool”的一节。如果另一个指向位置更合适，则可以选择另一个 HTML 页面。

配置 Kerberos 服务（任务）

本章介绍 KDC 服务器、网络应用服务器、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 321 页中的“配置 Kerberos 服务（任务列表）”
- 第 322 页中的“配置 KDC 服务器”
- 第 351 页中的“配置 Kerberos 客户机”
- 第 341 页中的“配置跨领域验证”
- 第 343 页中的“配置 Kerberos 网络应用服务器”
- 第 346 页中的“配置 Kerberos NFS 服务器”
- 第 367 页中的“同步 KDC 与 Kerberos 客户机的时钟”
- 第 368 页中的“交换主 KDC 服务器与从 KDC 服务器”
- 第 372 页中的“管理 Kerberos 数据库”
- 第 389 页中的“增强 Kerberos 服务器的安全性”

配置 Kerberos 服务（任务列表）

配置过程的某些部分依赖于其他部分，因此必须按特定顺序执行。这些过程通常用于建立使用 Kerberos 服务所需的服务。其他过程与顺序无关，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前解决配置问题。从长远角度来讲，提前规划能够节省您的时间和其他资源。	第 20 章，规划 Kerberos 服务

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 367 页中的“同步 KDC 与 Kerberos 客户机的时钟”
3. 配置 KDC 服务器。	为领域配置并构建主 KDC 服务器和从 KDC 服务器，以及 KDC 数据库。	第 322 页中的“配置 KDC 服务器”
4.（可选）增强 KDC 服务器的安全性。	防止 KDC 服务器中发生安全违规。	第 389 页中的“如何限制对 KDC 服务器的访问”
5.（可选）配置可交换 KDC 服务器。	使交换主 KDC 服务器与从 KDC 服务器的任务更轻松。	第 368 页中的“如何配置可交换的从 KDC 服务器”

配置额外 Kerberos 服务（任务列表）

完成必需的步骤后，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 341 页中的“配置跨领域验证”
配置 Kerberos 应用服务器。	使服务器支持使用 Kerberos 验证的服务，如 ftp、telnet 和 rsh。	第 343 页中的“配置 Kerberos 网络应用服务器”
配置 Kerberos 客户机。	使客户机使用 Kerberos 服务。	第 351 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 346 页中的“配置 Kerberos NFS 服务器”
增强应用服务器的安全性。	通过只允许访问通过验证的事务来增强应用服务器的安全性。	第 389 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 软件之后，必须配置 KDC 服务器。配置一个主 KDC 服务器和至少一个从 KDC 服务器可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 服务器与从 KDC 服务器之间最大的差别是，只有主 KDC 服务器可以处理数据库管理请求。例如，更改口令或添加新主体必须在主 KDC 服务器上完成。然后可以将这些更改传播到从 KDC 服务器。从 KDC 服务器和主 KDC 服务器都可以生成凭证。此功能可在主 KDC 服务器无法响应时提供冗余性。

表 21-1 配置 KDC 服务器（任务列表）

任务	说明	参考
配置主 KDC。	使用自动化过程为领域配置并构建主 KDC 服务器和数据库，这对于脚本来说非常有用。 使用交互式进程为领域配置并构建主 KDC 服务器和数据库，此种方法适用于大部分安装。 对于较为复杂的安装，需要使用手动过程为领域配置并构建主 KDC 服务器和数据库 通过使用手动过程并将 KDC 配置为使用 LDAP，为领域配置并构建主 KDC 服务器和数据库	第 323 页中的“如何自动配置主 KDC” 第 324 页中的“如何交互配置主 KDC” 第 325 页中的“如何手动配置主 KDC 服务器” 第 329 页中的“如何配置 KDC 以使用 LDAP 数据服务器”
配置从 KDC 服务器。	使用自动化过程为领域配置并构建从 KDC 服务器，这对于脚本来说非常有用。 使用交互式进程为领域配置并构建从 KDC 服务器，此种方法适用于大部分安装。 对于较为复杂的安装，需要使用手动过程为领域配置并构建从 KDC 服务器	第 335 页中的“如何自动配置从 KDC” 第 336 页中的“如何交互配置从 KDC” 第 337 页中的“如何手动配置从 KDC 服务器”
刷新 KDC 服务器上的主体密钥。	更新 KDC 服务器上的会话密钥，以使用新的加密类型。	第 340 页中的“如何刷新主服务器上的票证授予服务密钥”

▼ 如何自动配置主 KDC

在 Oracle Solaris 11 发行版中，使用下列过程可以自动配置主 KDC。

- 1 成为管理员，或者承担已指定给 Kerberos 服务器管理配置文件的角色或用户名。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 创建 KDC。

运行 kdcmgr 实用程序创建 KDC。您需要提供主密钥口令和管理主体的口令。

```
kdc1# kdcmgr -a kws/admin -r EXAMPLE.COM create master
```

```
Starting server setup
```

```
-----
```

```
Setting up /etc/krb5/kdc.conf
```

```
Setting up /etc/krb5/krb5.conf
```

```
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 password>
Re-enter KDC database master key to verify:  <Type it again>

Authenticating as principal root/admin@EXAMPLE.COM with password.
WARNING: no policy specified for kws/admin@EXAMPLE.COM; defaulting to no policy
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.

Setting up /etc/krb5/kadm5.acl.

-----
Setup COMPLETE.

kdc1#

```

▼ 如何交互配置主 KDC

在 Oracle Solaris 发行版中，使用下列过程可以交互配置主 KDC。

- 1 成为管理员，或者承担已指定给 Kerberos 服务器管理配置文件的角色或用户名。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。
- 2 创建 KDC。

运行 `kdcmgr` 实用程序创建 KDC。您需要提供主密钥口令和管理主体口令。

```

kdc1# kdcmgr create master

Starting server setup
-----

Enter the Kerberos realm: EXAMPLE.COM

Setting up /etc/krb5/kdc.conf

Setting up /etc/krb5/krb5.conf

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 password>
Re-enter KDC database master key to verify:  <Type it again>

Enter the krb5 administrative principal to be created: kws/admin

Authenticating as principal root/admin@EXAMPLE.COM with password.
WARNING: no policy specified for kws/admin@EXAMPLE.COM; defaulting to no policy
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.

Setting up /etc/krb5/kadm5.acl.

-----
Setup COMPLETE.

kdc1#

```

示例 21-1 显示 KDC 服务器的状态

kdcmgr status 命令可用于显示有关主或从 KDC 服务器的信息。

▼ 如何手动配置主 KDC 服务器

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://download.oracle.com/docs/cd/E23824_01/html/821-1456/aadmin-23.html

注 - 调整 URL 使其指向本部分，如第 320 页中的“图形化 Kerberos 管理工具中的联机帮助 URL”中所述。

开始之前 该过程要求将主机配置为使用 DNS。有关交换主 KDC 服务器的具体命名说明，请参见第 368 页中的“交换主 KDC 服务器与从 KDC 服务器”。

1 成为主 KDC 服务器上的超级用户。

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://download.oracle.com/docs/cd/E23824\_01/html/821-1456/aadmin-23.html
    }
```

在此示例中，`default_realm`、`kdc`、`admin_server` 行和所有 `domain_realm` 项都已被更改。此外还会编辑定义 `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。请参阅第 472 页中的“使用 Kerberos 加密类型”，了解关于限制加密类型的问题说明。

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
        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_uologsize = 1000
    }
```

在此示例中，`realms` 部分的领域名称定义被更改。而且，在 `realms` 部分，添加了用于启用增量传播和选择主 KDC 服务器将在日志中保留的更新数的行。

注 – 如果要限制加密类型，可以设置 `permitted_enctypes`、`supported_enctypes` 或 `master_key_type` 行。请参阅第 472 页中的“使用 Kerberos 加密类型”，了解关于限制加密类型的问题说明。

4 使用 `kdb5_util` 命令创建 KDC 数据库。

`kdb5_util` 命令可以创建 KDC 数据库。此外，使用 `-s` 选项时，该命令会在启动 `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 -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>
```

5 编辑 Kerberos 访问控制列表文件 (`kadm5.acl`)。

填充后，`/etc/krb5/kadm5.acl` 文件应包含所有获许管理 KDC 的主体名称。

```
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。缺省安装包括一个星号 (*)，以匹配所有的 `admin` 主体。此缺省安装可能会造成安全风险，因此，包括一个所有 `admin` 主体的列表会更安全。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。

接下来的子步骤会创建 Kerberos 服务使用的主体。

```
kdc1 # /usr/sbin/kadmin.local
kadmin.local:
```

a. 向数据库添加管理主体。

可以根据需要添加任意数目的 `admin` 主体。至少必须添加一个 `admin` 主体，这样才能完成 KDC 配置过程。此示例中添加了 `kws/admin` 主体。可以用适当的主体名称替代 `kws`。

```
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 服务器的更新。

```
kadmin.local: addprinc -randkey kiprop/kdc1.example.com
Principal "kiprop/kdc1.example.com@EXAMPLE.COM" created.
kadmin.local:
```

c. 退出 `kadmin.local`。

您已经为后续步骤添加了所有需要的主体。

```
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 并添加更多的主体。

此时，您可以使用图形化 Kerberos 管理工具添加主体。为此，必须使用之前在此过程中创建的一个 admin 主体名称登录。不过，为简便起见，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 服务器 host 主体。

基于 Kerberos 的应用程序（例如 kprop）将使用主机主体将变更传播到从 KDC 服务器。也可以通过该主体提供对使用应用程序（如 ssh）的 KDC 服务器的安全远程访问。请注意，当主体实例为主机名时，无论名称服务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. 可选创建 kclient 主体。

kclient 实用程序在 Kerberos 客户机安装期间使用该主体。如果您没有计划使用此实用程序，则不需要添加该主体。kclient 实用程序的用户需要使用该口令。

```
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 服务器的 host 主体添加到主 KDC 服务器的密钥表文件。

通过将主机主体添加到密钥表文件，允许应用服务器（如 sshd）自动使用该主体。

```
kadmin: ktadd host/kdc1.example.com
Entry for principal host/kdc1.example.com with kvno 3, encryption type AES-256 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 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 可选使用 NTP 或其他时钟同步机制同步主 KDC 服务器的时钟。

安装和使用网络时间协议 (Network Time Protocol, NTP) 并非必须的。但是，要成功进行验证，每个时钟必须遵守 `krb5.conf` 文件的 `libdefaults` 部分中定义的缺省时间。有关 NTP 的信息，请参见第 367 页中的“同步 KDC 与 Kerberos 客户机的时钟”。

10 配置从 KDC 服务器。

要提供冗余性，请确保至少安装了一个从 KDC 服务器。有关具体说明，请参见第 337 页中的“如何手动配置从 KDC 服务器”。

▼ 如何配置 KDC 以使用 LDAP 数据服务器

使用以下过程配置 KDC 以使用 LDAP 数据服务器。

此过程中使用以下配置参数：

- 领域名称 = `EXAMPLE.COM`
- DNS 域名 = `example.com`
- 主 KDC = `kdc1.example.com`
- 目录服务器 = `dsserver.example.com`
- admin 主体 = `kws/admin`
- LDAP 服务的 FMRI = `svc:/application/sun/ds:ds--var-opt-SUNWdsee-dsins1`
- 联机帮助 URL =
`http://download.oracle.com/docs/cd/E23824_01/html/821-1456/aadmin-23.html`

注 - 调整 URL 使其指向本部分，如第 320 页中的“图形化 Kerberos 管理工具中的联机帮助 URL”中所述。

开始之前 该过程也要求将主机配置为使用 DNS。为使性能更佳，请在同一台服务器上安装 KDC 和 LDAP 目录服务。此外还应运行目录服务器。下面的过程适用于运行 Sun Directory Server Enterprise Edition 7.0 发行版的服务器。

1 成为 KDC 上的超级用户。

2 配置主 KDC 以使用 SSL 访问目录服务器。

以下步骤将 Oracle Solaris 发行版 KDC 配置为使用 Directory Server 自签名证书。如果证书已过期，请按照“管理自签名证书”中的说明来续订证书。

a. 在目录服务器上，导出自签名目录服务器证书。

```
# /export/sun-ds6.1/ds6/bin/dsadm show-cert -F der /export/sun-ds6.1/directory2 \
defaultCert > /tmp/defaultCert.cert.der
```

b. 在主 KDC 上，导入目录服务器证书。

```
# pktool setpin keystore=nss dir=/var/ldap
# chmod a+r /var/ldap/*.db
# pktool import keystore=nss objtype=cert trust="CT" infile=/tmp/defaultCert.certutil.der \
  label=defaultCert dir=/var/ldap
```

c. 在主 KDC 上，测试 SSL 是否可以正常运行。

此示例假设 cn=directory manager 条目具有管理特权。

```
/usr/bin/ldapsearch -Z -P /var/ldap -D "cn=directory manager" \
  -h dsserver.example.com -b "" -s base objectclass='*'
Subject:
  "CN=dsserver.example.com,CN=636,CN=Directory Server,O=Example Corporation"
```

注意，CN=dsserver.example.com 条目应该包含全限定主机名，而不是简短版本。

3 如有需要，填充 LDAP 目录。

4 向现有架构中添加 Kerberos 架构。

```
# ldapmodify -h dsserver.example.com -D "cn=directory manager" -f /usr/share/lib/ldif/kerberos.ldif
```

5 在 LDAP 目录中创建 Kerberos 容器。

向 krb5.conf 文件中添加下列各项：

a. 定义数据库类型。

添加项来定义 realms 部分的 database_module。

```
database_module = LDAP
```

b. 定义数据库模块。

```
[dbmodules]
LDAP = {
    ldap_kerberos_container_dn = "cn=krbcontainer,dc=example,dc=com"
    db_library = kldap
    ldap_kdc_dn = "cn=kdc service,ou=profile,dc=example,dc=com"
    ldap_kadmin_dn = "cn=kadmin service,ou=profile,dc=example,dc=com"
    ldap_cert_path = /var/ldap
    ldap_servers = ldaps://dsserver.example.com
}
```

c. 在 LDAP 目录中创建 KDC。

以下命令创建 krbcontainer 及其他几个对象。还会创建 /var/krb5/.k5.EXAMPLE.COM 主密钥存储文件。

```
# kdb5_ldap_util -D "cn=directory manager" create -P abcd1234 -r EXAMPLE.COM -s
```

6 存储 KDC 绑定标识名 (Distinguished Name, DN) 口令。

当 KDC 绑定到 DS 时会使用这些口令。KDC 会根据其使用的访问类型使用不同的角色。

```
# kdb5_ldap_util stashesrvpw "cn=kdc service,ou=profile,dc=example,dc=com"
# kdb5_ldap_util stashesrvpw "cn=kadmin service,ou=profile,dc=example,dc=com"
```

7 添加 KDC 服务角色。

a. 创建一个内容如下的 `kdc_roles.ldif` 文件：

```
dn: cn=kdc service,ou=profile,dc=example,dc=com
cn: kdc service
sn: kdc service
objectclass: top
objectclass: person
userpassword: test123

dn: cn=kadmin service,ou=profile,dc=example,dc=com
cn: kadmin service
sn: kadmin service
objectclass: top
objectclass: person
userpassword: test123
```

b. 在 LDAP 目录中创建角色项

```
# ldapmodify -a -h dsserver.example.com -D "cn=directory manager" -f kdc_roles.ldif
```

8 为 KDC 相关角色设置 ACL。

```
# cat << EOF | ldapmodify -h dsserver.example.com -D "cn=directory manager"
# Set kadmin ACL for everything under krbcontainer.
dn: cn=krbcontainer,dc=example,dc=com
changetype: modify
add: aci
aci: (target="ldap:///cn=krbcontainer,dc=example,dc=com")(targetattr="krb*")(version 3.0;\
    acl kadmin_ACL; allow (all)\
    userdn = "ldap:///cn=kadmin service,ou=profile,dc=example,dc=com");)

# Set kadmin ACL for everything under the people subtree if there are
# mix-in entries for krb princis:
dn: ou=people,dc=example,dc=com
changetype: modify
add: aci
aci: (target="ldap:///ou=people,dc=example,dc=com")(targetattr="krb*")(version 3.0;\
    acl kadmin_ACL; allow (all)\
    userdn = "ldap:///cn=kadmin service,ou=profile,dc=example,dc=com");)
EOF
```

9 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://download.oracle.com/docs/cd/E23824_01/html/821-1456/aadmin-23.html
    }
```

在此示例中，default_realm、kdc、admin_server 行和所有 domain_realm 项都被更改。此外还会编辑定义 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。请参阅第 472 页中的“使用 Kerberos 加密类型”，了解关于限制加密类型的问题说明。

10 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
        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，realms 部分的领域名称定义被更改。而且，在 realms 部分，添加了用于启用增量传播和选择主 KDC 服务器将在日志中保留的更新数的行。

注 – 如果要限制加密类型，可以设置 `permitted_enctypes`、`supported_enctypes` 或 `master_key_type` 行。请参阅第 472 页中的“使用 Kerberos 加密类型”，了解关于限制加密类型的问题说明。

11 编辑 Kerberos 访问控制列表文件 (`kadm5.acl`)。

填充后，`/etc/krb5/kadm5.acl` 文件应包含所有获许管理 KDC 的主体名称。

```
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。缺省安装包括一个星号 (*)，以匹配所有的 `admin` 主体。此缺省安装可能会造成安全风险，因此，包括一个所有 `admin` 主体的列表会更安全。有关更多信息，请参见 [kadm5.acl\(4\)](#) 手册页。

12 启动 `kadmin.local` 命令并添加主体。

接下来的子步骤会创建 Kerberos 服务使用的主体。

```
kdc1 # /usr/sbin/kadmin.local
kadmin.local:
```

a. 向数据库添加管理主体。

可以根据需要添加任意数目的 `admin` 主体。至少必须添加一个 `admin` 主体，这样才能完成 KDC 配置过程。此示例中添加了 `kws/admin` 主体。可以用适当的主体名称替代 `kws`。

```
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. 退出 `kadmin.local`。

您已经为后续步骤添加了所有需要的主体。

```
kadmin.local: quit
```

13 可选配置 Kerberos 服务的 LDAP 相关项。

如果 LDAP 和 KDC 服务器运行在同一台主机上，且 LDAP 服务以 SMF FMRI 配置，则添加 Kerberos 守护进程对 LDAP 服务的依赖性。如果 LDAP 服务重新启动，此依赖性将重新启动 KDC 服务。

a. 添加对 `krb5kdc` 服务的依赖性。

```
# svccfg -s security/krb5kdc
svc:/network/security/krb5kdc> addpg dsins1 dependency
svc:/network/security/krb5kdc> setprop dsins1/entities = \
    fmri: "svc:/application/sun/ds:ds--var-opt-SUNWdsee-dsins1"
svc:/network/security/krb5kdc> setprop dsins1/grouping = astring: "require_all"
svc:/network/security/krb5kdc> setprop dsins1/restart_on = astring: "restart"
```

```
svc:/network/security/krb5kdc> setprop dsins1/type = astring: "service"
svc:/network/security/krb5kdc> exit
```

b. 添加对 kadmin 服务的依赖性。

```
# svccfg -s security/kadmin
svc:/network/security/kadmin> addpg dsins1 dependency
svc:/network/security/kadmin> setprop dsins1/entities =\
    fmri: "svc:/application/sun/ds:ds--var-opt-SUNWdsee-dsins1"
svc:/network/security/kadmin> setprop dsins1/grouping = astring: "require_all"
svc:/network/security/kadmin> setprop dsins1/restart_on = astring: "restart"
svc:/network/security/kadmin> setprop dsins1/type = astring: "service"
svc:/network/security/kadmin> exit
```

14 启动 Kerberos 守护进程。

```
kdc1 # svcadm enable -r network/security/krb5kdc
kdc1 # svcadm enable -r network/security/kadmin
```

15 启动 kadmin 并添加更多的主体。

此时，您可以使用 GUI Kerberos 管理工具添加主体。为此，必须使用之前在此过程中创建的一个 admin 主体名称登录。不过，为简便起见，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 服务器 host 主体。

基于 Kerberos 的应用程序（如 klist 和 kprop）使用 host 主体。挂载通过验证的 NFS 文件系统时，客户机使用该主体。请注意，当主体实例为主机名时，无论名称服务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. 可选创建 kclient 主体。

kclient 实用程序在 Kerberos 客户机安装期间使用该主体。如果您没有计划使用此实用程序，则不需要添加该主体。kclient 实用程序的用户需要使用该口令。

```
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 服务器的 host 主体添加到主 KDC 服务器的密钥表文件。

将主机主体添加到密钥表文件会允许自动使用该主体。

```
kadmin: ktadd host/kdc1.example.com
Entry for principal host/kdc1.example.com with kvno 3, encryption type AES-256 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 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
```

16 可选使用 NTP 或其他时钟同步机制同步主 KDC 服务器的时钟。

安装和使用网络时间协议 (Network Time Protocol, NTP) 并非必须的。但是，要成功进行验证，每个时钟必须遵守 `krb5.conf` 文件的 `libdefaults` 部分中定义的缺省时间。有关 NTP 的信息，请参见第 367 页中的“同步 KDC 与 Kerberos 客户机的时钟”。

17 配置从 KDC 服务器。

要提供冗余性，请确保至少安装了一个从 KDC 服务器。有关具体说明，请参见第 337 页中的“如何手动配置从 KDC 服务器”。

▼ 如何自动配置从 KDC

在 Oracle Solaris 发行版中，使用下列过程可以自动配置从 KDC。

1 成为管理员，或者承担已指定给 Kerberos 服务器管理配置文件的角色或用户名。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 创建 KDC。

运行 `kdcmgr` 实用程序创建 KDC。您需要提供主密钥口令和管理主体的口令。

```
kdc2# kdcmgr -a kws/admin -r EXAMPLE.COM create -m kdc1 slave
```

```
Starting server setup
```

```
-----
```

```
Setting up /etc/krb5/kdc.conf
```

```
Setting up /etc/krb5/krb5.conf
```

```
Obtaining TGT for kws/admin ...
```

```
Password for kws/admin@EXAMPLE.COM: <Type the password>
```

```
Setting up /etc/krb5/kadm5.acl.
```

```
Setting up /etc/krb5/kpropd.acl.
```

```
Waiting for database from master...
```

```
Waiting for database from master...
```

```

Waiting for database from master...
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 password>

-----
Setup COMPLETE.

kdc2#

```

▼ 如何交互配置从 KDC

使用以下过程交互配置从 KDC。

- 1 成为管理员，或者承担已指定给 Kerberos 服务器管理配置文件的角色或用户名。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。
- 2 创建 KDC。

运行 `kdcmgr` 实用程序创建 KDC。您需要提供主密钥口令和管理主体的口令。

```

kdc1# kdcmgr create slave

Starting server setup
-----

Enter the Kerberos realm: EXAMPLE.COM
What is the master KDC's host name?: kdc1

Setting up /etc/krb5/kdc.conf

Setting up /etc/krb5/krb5.conf
Obtaining TGT for kws/admin ...
Password for kws/admin@EXAMPLE.COM:      <Type the password>

Setting up /etc/krb5/kadm5.acl.

Setting up /etc/krb5/kpropd.acl.

Waiting for database from master...
Waiting for database from master...
Waiting for database from master...
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 password>

-----
Setup COMPLETE.

kdc2#

```

▼ 如何手动配置从 KDC 服务器

在此过程中，将配置新的名为 kdc2 的从 KDC 服务器。此外还会配置增量传播。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

开始之前 必须配置主 KDC 服务器。要了解交换从 KDC 服务器的具体说明，请参见第 368 页中的“[交换主 KDC 服务器与从 KDC 服务器](#)”。

1 成为主 KDC 服务器上的超级用户。

2 在主 KDC 服务器上，启动 kadmin。

必须使用在配置主 KDC 服务器时创建的一个 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 服务器上，向数据库中添加从主机主体（如果尚未完成）。

要使从 KDC 服务器正常工作，它必须有主机主体。请注意，当主体实例为主机名时，无论名称服务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@EXAMPLE.COM" created.
kadmin:
```

b. 在主 KDC 服务器上，创建 kiprof 主体。

kiprof 主体用于授权来自主 KDC 服务器的增量传播。

```
kadmin: addprinc -randkey kiprof/kdc2.example.com
Principal "kiprof/kdc2.example.com@EXAMPLE.COM" created.
kadmin:
```

c. 退出 kadmin。

```
kadmin: quit
```

3 在主 KDC 服务器上，编辑 Kerberos 配置文件 (krb5.conf)。

需要为每个从 KDC 服务器添加一个项。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 服务器上，向 **kadm5.acl** 添加 **kiprop** 项。

通过此项，主 KDC 服务器可以接收对 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 服务器上，重新启动 **kadmind** 以使用 **kadm5.acl** 文件中新的项。

```

kdc1 # svcadm restart network/security/kadmin

```

- 6 在所有的从 KDC 服务器上，从主 KDC 服务器复制 KDC 管理文件。

需要在所有的从 KDC 服务器上执行该步骤，因为主 KDC 服务器已经更新了每个 KDC 服务器需要的信息。可以使用 **ftp** 或类似的传输机制从主 KDC 服务器获取以下文件的副本：

- /etc/krb5/krb5.conf
- /etc/krb5/kdc.conf

- 7 在所有的从 KDC 服务器上，将主 KDC 服务器和每个从 KDC 服务器的项添加到数据库传播配置文件 **kpropd.acl** 中。

需要在所有从 KDC 服务器上更新此信息。

```

kdc2 # cat /etc/krb5/kpropd.acl
host/kdc1.example.com@EXAMPLE.COM
host/kdc2.example.com@EXAMPLE.COM

```

- 8 在所有的从 KDC 服务器上，确保未填充 Kerberos 访问控制列表文件 **kadm5.acl**。

未修改的 **kadm5.acl** 文件类似如下示例：

```

kdc2 # cat /etc/krb5/kadm5.acl
*/admin@__default_realm__ *

```

如果该文件中包含 **kiprop** 项，请将其删除。

- 9 在新的从 KDC 服务器上，更改 **kdc.conf** 中的某个项。

使用定义 **sunw_dbprop_slave_poll** 的项替换 **sunw_dbprop_master_uologsize** 项。该项将轮询时间设为两分钟。

```

kdc1 # cat /etc/krb5/kdc.conf
[kdcdefaults]
    kdc_ports = 88,750

[realms]
    EXAMPLE.COM= {
        profile = /etc/krb5/krb5.conf
        database_name = /var/krb5/principal
    }

```

```

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 服务器上，启动 **kadmin** 命令。

必须使用在配置主 KDC 服务器时创建的一个 **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 服务器的主机主体添加到从 KDC 服务器的密钥表文件中。

该项可使 **kprop** 及其他基于 Kerberos 的应用程序正常工作。请注意，当主体实例为主机名时，无论名称服务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-256 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 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 **kprop** 主体添加到从 KDC 服务器的密钥表文件中。

通过将 **kprop** 主体添加到 **krb5.keytab** 文件，允许 **kpropd** 命令在启动增量传播时对其自身进行验证。

```

kadmin: ktadd kprop/kdc2.example.com
Entry for principal kprop/kdc2.example.com with kvno 3, encryption type AES-256 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal kprop/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 kprop/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 kprop/kdc2.example.com with kvno 3, encryption type ArcFour
with HMAC/md5 added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal kprop/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 服务器上，启动 Kerberos 传播守护进程。

```
kdc2 # svcadm enable network/security/krb5_prop
```

- 12 在新的从 KDC 服务器上，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 可选在新的从 KDC 服务器上，使用 NTP 或其他时钟同步机制同步主 KDC 服务器时钟。

安装和使用网络时间协议 (Network Time Protocol, NTP) 并非必须的。但是，要成功进行验证，每个时钟必须遵守 `krb5.conf` 文件的 `libdefaults` 部分中定义的缺省时间。有关 NTP 的信息，请参见第 367 页中的“同步 KDC 与 Kerberos 客户机的时钟”。

- 14 在新的从 KDC 服务器上，启动 KDC 守护进程 (`krb5kdc`)。

```
kdc2 # svcadm enable network/security/krb5kdc
```

▼ 如何刷新主服务器上的票证授予服务密钥

当票证授予服务 (Ticket Granting Service, TGS) 主体仅具有 DES 密钥时（在 Solaris 10 发行版之前创建的 KDC 服务器即为这种情况），该密钥会将票证授予票证 (Ticket Granting Ticket, TGT) 会话密钥的加密类型限制为 DES。如果某个 KDC 更新到支持其他更强加密类型的发行版，管理员可能会希望为该 KDC 生成的所有会话密钥使用这种更强的加密。但是，如果现有的 TGS 主体没有刷新其密钥使之包括新的加密类型，TGT 会话密钥仍会继续限制为 DES。以下过程会刷新密钥，以便可以使用其他的加密类型。

- 刷新 TGS 服务主体密钥。

```
kdc1 % /usr/sbin/kadmin -p kws/admin
Enter password:      <Type kws/admin password>
kadmin: cpw -randkey krbtgt/EXAMPLE.COM@EXAMPLE.COM
```

示例 21-2 从主服务器刷新主体密钥

如果以 `root` 身份登录到 KDC 主服务器，您可以通过以下命令刷新 TGS 服务主体：

```
kdc1 # kadmin.local -q 'cpw -randkey krbtgt/EXAMPLE.COM@EXAMPLE.COM'
```


配置跨领域验证

有几种方法可以将各个领域链接在一起，从而允许在一个领域中验证另一个领域中的用户。跨领域验证通过建立一个由两个领域共享的密钥来实现。领域之间的关系可以是分层关系或直接关系（请参见第 314 页中的“领域分层结构”）。

▼ 如何建立层次化跨领域验证

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。必须在主 KDC 服务器上在这两个领域中完成此过程。

开始之前 必须配置每个领域的主 KDC 服务器。要完全测试验证过程，必须配置多个 Kerberos 客户机。

1 成为第一个主 KDC 服务器上的超级用户。

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 服务器时创建的一个 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完全相同。因此，服务主体 krbtgt/ENG.EAST.EXAMPLE.COM@EAST.EXAMPLE.COM 的口令在两个领域中必须相同。

3 向 Kerberos 配置文件 (krb5.conf) 中添加项来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该文件，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 服务器和其他服务器）都必须安装了 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。

▼ 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。必须在主 KDC 服务器上在这两个领域中完成此过程。

开始之前 必须配置每个领域的主 KDC 服务器。要完全测试验证过程，必须配置多个 Kerberos 客户机。

- 1 成为一个主 KDC 服务器上的超级用户。

- 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 服务器时创建的一个 `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
krtgt/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
krtgt/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完全相同。因此，服务主体 `krbtgt/ENG.EAST.EXAMPLE.COM@SALES.WEST.EXAMPLE.COM` 的口令在两个领域中必须相同。

- 3 向 Kerberos 配置文件中添加项来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` 领域中的适当定义。

```
# 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 服务器和其他服务器）都必须安装了 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 网络应用服务器

网络应用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、ssh 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 网络应用服务器

此过程使用以下配置参数：

- 应用服务器 = 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 服务器。要完全测试该过程，必须配置多个 Kerberos 客户机。

- 1 成为服务器上的超级用户。
- 2 可选安装 NTP 客户机或其他时钟同步机制。

有关 NTP 的信息，请参见第 367 页中的“同步 KDC 与 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文件。

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，请执行以下步骤创建新主体。

有关如何使用 GUI Kerberos 管理工具添加主体的说明，请参见第 421 页中的“[如何创建新的 Kerberos 主体](#)”。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 服务器时创建的一个 `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。

按照下列方法使用 `host` 主体：

- 使用远程命令时（如 `rsh` 和 `ssh`）时验证通信。
- 供 `pam_krb5` 防止 KDC 欺骗攻击，以确认用户的 Kerberos 凭证是从可信 KDC 获取的。
- 允许 `root` 用户在不要求存在 `root` 主体的情况下，自动获取 Kerberos 凭证。在执行手动 NFS 挂载时，若共享需要使用 Kerberos 凭证，此功能很有用。

如果要使用 Kerberos 服务验证使用远程应用程序的通信，则需要该主体。如果服务器有多个与之关联的主机名，则应使用主机名的 FQDN 格式为每个主机名创建一个主体。

```
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文件中。

如果没有运行 `kadmin` 命令，请使用以下类似语法重新启动该命令：`/usr/sbin/kadmin -p kws/admin`

如果服务器有多个与之相关的主机名，请为每个主机名向密钥表添加一个主体。

```
kadmin: ktadd host/boston.example.com
Entry for principal host/boston.example.com with kvno 3, encryption type AES-256 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 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
```

▼ 运行 FTP 时如何使用带有 Kerberos 的通用安全服务

可将通用安全服务 (generic security service, GSS) 应用于应用程序中，以轻松使用 Kerberos 实现验证、完整性和保密性。以下步骤显示如何为 ProFTPD 启用 GSS。

- 1 成为 FTP 服务器上的超级用户。
- 2 为 FTP 服务器添加主体并更新该服务器的密钥表文件。

如果您之前执行过更改，可能无需执行下列步骤。

- a. 启动 kadmin 命令。

```
ftpserver1 # /usr/sbin/kadmin -p kws/admin
Enter password:      <Type kws/admin password>
kadmin:
```

- b. 为 FTP 服务器添加 host 服务主体。

```
kadmin: addprinc -randkey host/ftpserver1.example.com
```

- c. 将 host 服务主体添加到服务器的 keytab 文件中。

```
kadmin: ktadd host/ftpserver1.example.com
```

- 3 为 FTP 服务器启用 GSS。

对 /etc/proftpd.conf 文件执行下列更改。

```
# cat /etc/proftpd.conf
#User          ftp
#Group         ftp

User           root
Group          root

UseIPv6 off

LoadModule     mod_gss.c

GSSEngine      on
GSSKeytab      /etc/krb5/krb5.keytab
```

- 4 重新启动 FTP 服务器。

```
# svcadm restart network/ftp
```

配置 Kerberos NFS 服务器

NFS 服务使用 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 317 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 NFS 挂载的文件系统启动 Kerberos 安全模式所需执行的任务。以下任务列表描述了本节所包含的任务。

表 21-2 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 346 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 348 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 348 页中的“如何向凭证表中添加单个项”
创建两个类似领域之间的凭证映射。	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 349 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 350 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

此过程中使用以下配置参数：

- 领域名称 = EXAMPLE.COM
- DNS 域名 = example.com
- NFS 服务器 = denver.example.com
- admin 主体 = kws/admin

- 1 成为 NFS 服务器上的超级用户。
- 2 完成配置 Kerberos NFS 服务器的先决条件。
必须配置主 KDC 服务器。要完全测试此过程，需要多个客户机。

3 可选安装 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) 并非必须的。但是，要成功进行验证，每个时钟必须与 KDC 服务器上的时间同步，最大时差不超过 `krb5.conf` 文件中 `clockskew` 关系定义的值。有关 NTP 的信息，请参见第 367 页中的“同步 KDC 与 Kerberos 客户机的时钟”。

4 将 NFS 服务器配置为 Kerberos 客户机。

请按照第 351 页中的“配置 Kerberos 客户机”中的说明操作。

5 启动 `kadmin`。

可以使用图形化 Kerberos 管理工具添加主体，如第 421 页中的“如何创建新的 Kerberos 主体”所述。为此，必须使用在配置主 KDC 服务器时创建的一个 `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，当主体实例为主机名时，无论名称服务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都具有唯一名称，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-256 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 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
```

6 可选如果需要，创建特殊 GSS 凭证映射。

通常，Kerberos 服务生成 GSS 凭证与 UNIX UID 之间的适当映射。缺省映射在[第 317 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 348 页](#)中的“如何创建凭证表”了解更多信息。

7 使用 Kerberos 安全模式共享 NFS 文件系统。

有关更多信息，请参见[第 350 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。缺省情况下，主体名称的基本部分与 UNIX 登录名匹配。要使 NFS 客户机通过 Kerberos 验证从 NFS 服务器挂载文件系统，如果缺省映射不满足要求，则必须创建此表。

1 成为 NFS 服务器上的超级用户。

2 编辑 `/etc/gss/gsscred.conf` 并更改安全机制。

将机制更改为 `files`。

3 使用 `gsscred` 命令创建凭证表。

```
# gsscred -m kerberos_v5 -a
```

`gsscred` 命令从 `svc:/system/name-service/switch:default` 服务的 `passwd` 项列出的所有源中收集信息。如果您希望凭证表中不包含本地口令项，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 服务器上创建了 `gsscred` 表。有关说明，请参见[第 348 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一个项。

```
# 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 将 UID 添加到主体名称映射。

示例 21-3 向凭证表中添加多组成部分主体

在以下示例中，将为名为 `sandy/admin` 的主体添加一个项，该主体映射到 UID 3736。

```
# gsscred -m kerberos_v5 -n sandy/admin -u 3736 -a
```

示例 21-4 向凭证表中添加其他域中的主体

在以下示例中，将为名为 `sandy/admin@EXAMPLE.COM` 的主体添加一个项，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

此过程提供使用同一个口令文件的领域之间的正确凭证映射。在此示例中，领域 `CORP.EXAMPLE.COM` 和 `SALES.EXAMPLE.COM` 使用同一个口令文件。`bob@CORP.EXAMPLE.COM` 和 `bob@SALES.EXAMPLE.COM` 的凭证映射到同一个 UID。

- 1 成为客户机系统上的超级用户。
- 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
        .
    }
```

示例 21-5 在使用同一个口令文件的各领域之间映射凭证

此示例提供使用同一个口令文件的领域之间的正确凭证映射。在此示例中，领域 `CORP.EXAMPLE.COM` 和 `SALES.EXAMPLE.COM` 使用同一个口令文件。`bob@CORP.EXAMPLE.COM` 和 `bob@SALES.EXAMPLE.COM` 的凭证映射到同一个 UID。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 有关凭证映射问题故障排除过程的帮助信息，请参见第 408 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方式提供安全的 NFS 访问。客户机与 NFS 服务器协商安全模式时，将使用该客户机有权访问的服务器所提供的第一种模式。此模式用于该 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 检验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，则您需要确认是否已完成第 346 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 使用适当的安全模式共享文件系统。

```
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需要进行 Kerberos 验证。要访问文件，应验证 NFS 客户机上的用户主体。

- 5 可选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可以接受安全模式的缺省选择，则无需执行此过程。

```
file-system auto_home -nosuid,sec=mode
```

- 6 可选手动发出 **mount** 命令以使用非缺省模式访问文件系统。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 21-6 使用一种 Kerberos 安全模式共享文件系统

在此示例中，Kerberos 验证必须成功，才能通过 NFS 服务访问文件。

```
# share -F nfs -o sec=krb5 /export/home
```

示例 21-7 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。客户机与 NFS 服务器协商决定使用哪种模式。如果命令中的第一种模式失败，即将尝试下一种。有关更多信息，请参见 [nfssec\(5\)](#) 手册页。

```
# share -F nfs -o sec=krb5:krb5i:krb5p /export/home
```

配置 Kerberos 客户机

Kerberos 客户机包括网络上需要使用 Kerberos 服务的所有主机（不是 KDC 服务器）。本节介绍有关安装 Kerberos 客户机的过程，以及关于使用 **root** 验证来挂载 NFS 文件系统的特定信息。

配置 Kerberos 客户机（任务列表）

以下任务列表涵盖有关设置 Kerberos 客户机的所有过程。每行都包含一个任务、关于执行该项任务的原因说明，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 352 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。如果每个客户机安装都要求唯一的安装参数，请使用此过程。 自动安装 Kerberos 客户机。如果每个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。如果仅需更改少数几个安装参数，请使用此过程。 自动安装 Active Directory 服务器的 Kerberos 客户机。	第 358 页中的“如何手动配置 Kerberos 客户机” 第 353 页中的“如何自动配置 Kerberos 客户机” 第 354 页中的“如何交互配置 Kerberos 客户机” 第 357 页中的“如何为 Active Directory 服务器配置 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，还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 363 页中的“如何以 root 用户身份访问受 Kerberos 保护的 NFS 文件系统”
禁用颁发客户机票证授予票证 (Ticket Granting Ticket, TGT) 的 KDC 的验证。	允许在本地密钥表文件中没有存储主机主体的客户机跳过目的在于验证颁发 TGT 的 KDC 是否为颁发主机主体的服务器的安全检查。	第 362 页中的“如何禁用票务授予票证的验证”

▼ 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 配置文件可以降低键入错误的可能性。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/denver.example.com/export/install/profile
REALM EXAMPLE.COM
KDC kdc1.example.com
ADMIN clntconfig
FILEPATH /net/denver.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 352 页中的“如何创建 Kerberos 客户机安装配置文件”。

- 1 成为管理员，或者承担已指定给 Kerberos 客户机管理配置文件的角色或用户名。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。

- 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/denver.example.com/export/install/profile
```

```
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/denver.example.com/export/install/krb5.conf.
```

```
-----
```

```
Setup COMPLETE.
```

```
client#
```

示例 21-8 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/denver.example.com/export/install/profile\
-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/denver.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` 安装实用程序而非安装配置文件。在 Oracle Solaris 11 发行版中，`kclient` 实用程序不仅提高了易用性，也提高了与 Active Directory 服务器协作的能力。有关更多信息，请参见第 357 页中的“[如何为 Active Directory 服务器配置 Kerberos 客户机](#)”。有关在早期版本上运行 `kclient` 的示例，请参见示例 21-10。

- 1 成为管理员，或者承担已指定给 Kerberos 客户机管理配置文件的角色或用户名。
有关更多信息，请参见第 150 页中的“[如何获取管理权限](#)”。

- 2 运行 `kclient` 安装脚本。

需要提供以下信息：

- Kerberos 领域名称
- 主 KDC 服务器主机名
- KDC 从主机名
- 映射到本地领域的域
- 用于 Kerberos 验证的 PAM 服务名称和选项

- a. 指示 KDC 服务器是否运行 Oracle Solaris 发行版。

如果此系统是一个未运行 Oracle Solaris 发行版的 KDC 服务器的客户机，则需要定义运行 KDC 的服务器类型。可用的服务器有：Microsoft Active Directory、MIT KDC 服务器、Heimdal KDC 服务器和 Shishi KDC 服务器。

- b. 选择是否将 DNS 用于 Kerberos 查找。

如果将 DNS 用于 Kerberos 查找，需要输入您希望使用的 DNS 查找选项。有效的选项有：`dns_lookup_kdc`、`dns_lookup_realm` 和 `dns_fallback`。有关这些值的更多信息，请参见 `krb5.conf(4)` 手册页。

- c. 定义 Kerberos 区域的名称和主 KDC 主机名。
此信息将添加到 `/etc/krb5/krb5.conf` 配置文件中。
- d. 选择从 KDC 是否存在。
如果区域中存在从 KDC，则需要输入从 KDC 主机名。此信息用于在客户机配置文件中创建其他 KDC 项。
- e. 指示是否需要服务或主机密钥。
通常情况下不需要服务和主机密钥，除非客户机系统正在托管基于 Kerberos 的服务。
- f. 指定客户机是否是群集的成员。
如果客户机是群集的成员，您需要提供该群集的逻辑名称。逻辑主机名可用于创建服务密钥，而后者是在群集中托管 Kerberos 服务所必需的。
- g. 确定映射到当前领域的所有域和主机。
此映射允许其他域存在于客户机的缺省领域。
- h. 指定客户机是否会使用基于 Kerberos 的 NFS。
当客户机使用 Kerberos 托管 NFS 服务时，需要创建 NFS 服务密钥。
- i. 指示是否需要更新 `/etc/pam.conf` 文件。
这样，您可以设置哪个 PAM 服务使用 Kerberos 进行验证。您需要输入服务名称，以及一个标志来指示如何使用 Kerberos 验证。有效的标志选项包括：
 - `first`—首先使用 Kerberos 验证，并且当 Kerberos 验证失败时仅使用 UNIX
 - `only`—仅使用 Kerberos 验证
 - `optional`—有选择的使用 Kerberos 验证
- j. 选择是否复制主 `/etc/krb5/krb5.conf` 文件。
借助此选项，可以指定在 `kclient` 的参数不足时，使用特定的配置信息。

示例 21-9 运行 kclient 安装实用程序

```
client# /usr/sbin/kclient

Starting client setup
-----

Is this a client of a non-Solaris KDC ? [y/n]: n
    No action performed.
Do you want to use DNS for kerveros lookups ? [y/n]: n
    No action performed.
Enter the Kerberos realm: EXAMPLE.COM
Specify the KDC hostname for the above realm: kdc1.example.com
```

Note, this system and the KDC's time must be within 5 minutes of each other for Kerberos to function. Both systems should run some form of time synchronization system like Network Time Protocol (NTP).

Do you have any slave KDC(s) ? [y/n]: **y**
Enter a comma-separated list of slave KDC host names: **kdc2.example.com**

Will this client need service keys ? [y/n]: **n**
No action performed.

Is this client a member of a cluster that uses a logical host name ? [y/n]: **n**
No action performed.

Do you have multiple domains/hosts to map to realm ? [y/n]: **y**
Enter a comma-separated list of domain/hosts to map to the default realm: **engineering.example.com, \example.com**

Setting up /etc/krb5/krb5.conf.

Do you plan on doing Kerberized nfs ? [y/n]: **y**
Do you want to update /etc/pam.conf ? [y/n]: **y**
Enter a comma-separated list of PAM service names in the following format:
service:{first|only|optional}: **xscreensaver:first**
Configuring /etc/pam.conf.

Do you want to copy over the master krb5.conf file ? [y/n]: **n**
No action performed.

Setup COMPLETE.

示例 21-10 在 Oracle Solaris 10 发行版中运行 kclient 安装实用程序

以下输出显示了 kclient 命令的运行结果。

```
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/denver.example.com/export/install/krb5.conf

Copied /net/denver.example.com/export/install/krb5.conf.

-----
Setup COMPLETE !
#
```

▼ 如何为 Active Directory 服务器配置 Kerberos 客户端

此过程使用 `kclient` 安装实用程序而非安装配置文件。

1 成为超级用户。

2 可选启用针对客户端的 DNS 资源记录创建。

```
client# sharectl set -p ddns_enable=true smb
```

3 运行 `kclient` 实用程序。

`-T` 选项表示对 KDC 服务器类型的选择。在本例中，选择的是 Active Directory 服务器。

```
client# kclient -T ms_ad
```

缺省情况下，需要提供管理员主体的口令。

示例 21-11 使用 `kclient` 配置适用于 Active Directory 服务器的 Kerberos 客户端

以下输出显示了使用 `ms_ad` (Microsoft Active Directory) 服务器类型参数运行 `kclient` 命令的结果。此客户端将加入名为 `EXAMPLE.COM` 的 Active Directory 域。

```
client# /usr/sbin/kclient -T ms_ad

Starting client setup
-----

Attempting to join 'CLIENT' to the 'EXAMPLE.COM' domain.
Password for Administrator@EXAMPLE.COM: <Type the password>
Forest name found: example.com
Looking for local KDCs, DCs and global catalog servers (SVR RRs).

Setting up /etc/krb5/krb5.conf

Creating the machine account in AD via LDAP.
-----
Setup COMPLETE.
#
```

▼ 如何手动配置 Kerberos 客户机

此过程中使用以下配置参数：

- 领域名称 = EXAMPLE.COM
- DNS 域名 = example.com
- 主 KDC = kdc1.example.com
- 从 KDC = kdc2.example.com
- NFS 服务器 = denver.example.com
- 客户机 = client.example.com
- admin 主体 = kws/admin
- 用户主体 = mre
- 联机帮助 URL =
http://download.oracle.com/docs/cd/E23824_01/html/821-1456/aadmin-23.html

注 – 调整 URL 使其指向本部分，如第 320 页中的“图形化 Kerberos 管理工具中的联机帮助 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。还需要指定 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://download.oracle.com/docs/cd/E23824\_01/html/821-1456/aadmin-23.html
```

注 – 如果要限制加密类型，可以设置 `default_tkt_enctypes` 或 `default_tgs_enctypes` 行。请参阅第 472 页中的“使用 Kerberos 加密类型”，了解关于限制加密类型的问题说明。

3 可选更改用于定位 KDC 的过程。

缺省情况下，Kerberos 领域到 KDC 的映射按如下顺序确定：

- `krb5.conf` 中 `realms` 部分的定义
- 通过在 DNS 中查找 SRV 记录。

将 `dns_lookup_kdc` 或 `dns_fallback` 添加到 `krb5.conf` 文件的 `libdefaults` 部分，可以更改此行为。有关更多信息，请参见 [krb5.conf\(4\)](#) 手册页。请注意，始终会首先尝试引用。

4 可选更改用于确定主机领域的过程。

缺省情况下，主机到领域的映射按如下顺序确定：

- 如果 KDC 支持引用，则 KDC 会通知客户机该主机属于哪个领域。
- `krb5.conf` 文件中的 `domain_realm` 定义。
- 主机的 DNS 域名。
- 缺省领域。

将 `dns_lookup_kdc` 或 `dns_fallback` 添加到 `krb5.conf` 文件的 `libdefaults` 部分，可以更改此行为。有关更多信息，请参见 [krb5.conf\(4\)](#) 手册页。请注意，始终会首先尝试引用。

5 可选使用 NTP 或其他时钟同步机制将客户机时钟与主 KDC 服务器时钟同步。

安装和使用网络时间协议 (Network Time Protocol, NTP) 并非必须的。但是，要成功进行验证，每个时钟必须与 KDC 服务器上的时间同步，最大时差不超过 `krb5.conf` 文件中 `clockskew` 关系定义的值。有关 NTP 的信息，请参见第 367 页中的“同步 KDC 与 Kerberos 客户机的时钟”。

6 启动 `kadmin`。

可以使用图形化 Kerberos 管理工具添加主体，如第 421 页中的“如何创建新的 Kerberos 主体”所述。为此，必须使用在配置主 KDC 服务器时创建的一个 `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. 可选如果不存在用户主体，请创建一个。

仅当尚未对与该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. 可选创建 root 主体，并将其添加到服务器的密钥表文件。

如果客户机需要具有对使用 NFS 服务挂载的远程文件系统的 root 访问权限，则必须执行此步骤。如果需要非交互式 root 访问权限（如以 root 身份运行 cron 作业），也必须执行该步骤。

如果客户机不需要对使用 NFS 服务挂载的远程文件系统的 root 访问权限，则可以跳过该步骤。root 主体应有两个组成部分，第二个部分是 Kerberos 客户机系统的主机名，以避免创建领域范围的 root 主体。请注意，当主体实例为主机名时，无论名称服务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: ktadd root/client.example.com
Entry for principal root/client.example.com with kvno 3, encryption type AES-256 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 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. 创建 root 主体，并将其添加到服务器的密钥表文件。

远程访问服务使用 host 主体来提供验证。如果密钥表文件中尚无凭证，此主体允许 root 获取凭证。

```
kadmin: addprinc -randkey host/denver.example.com
Principal "host/denver.example.com@EXAMPLE.COM" created.
kadmin: ktadd host/denver.example.com
Entry for principal host/denver.example.com with kvno 3, encryption type AES-256 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 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:
```

d. 可选将服务器的 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该步骤。

```
kadmin: ktadd nfs/denver.example.com
Entry for principal nfs/denver.example.com with kvno 3, encryption type AES-256 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 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. 退出 kadmin。

```
kadmin: quit
```

7 可选对 NFS 启用 Kerberos。

a. 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

```

b. 启用 DNS。

如未启用 `svc:/network/dns/client:default` 服务，请启用它。有关更多信息，请参见 [resolv.conf\(4\)](#) 手册页。

c. 重新启动 gssd 服务。

```
# svcadm restart network/rpc/gss
```

8 如果您希望客户机自动续订 TGT 或警告用户 Kerberos 票证失效，请在 /etc/krb5/warn.conf 文件中创建一个相应项。

有关更多信息，请参见 [warn.conf\(4\)](#) 手册页。

示例 21-12 使用非 Solaris KDC 设置 Kerberos 客户机

Kerberos 客户机可以设置为与非 Solaris KDC 一起工作。在这种情况下，必须在 /etc/krb5/krb5.conf 文件的 `realms` 部分中包括一行。该行会更改客户机与 Kerberos 口令更改服务器通信时使用的协议。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 21-13 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

kdc1          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 21-14 Kerberos 服务器位置的 DNS SRV 记录

此示例分别为 KDC、admin 服务器和 kpasswd 服务器的位置定义记录。

```
@ IN SOA kdc1.example.com root.kdc1.example.com (
                                1989020501 ;serial
                                10800      ;refresh
                                3600       ;retry
                                3600000    ;expire
                                86400     ) ;minimum

kdc1          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 kdc2.example.com
_kerberos._tcp.EXAMPLE.COM      IN      SRV  0 0 88 kdc2.example.com
_kerberos._udp.EXAMPLE.COM      IN      SRV  1 0 88 kdc1.example.com
_kerberos._tcp.EXAMPLE.COM      IN      SRV  1 0 88 kdc1.example.com
_kerberos-admin._tcp.EXAMPLE.COM IN      SRV  0 0 749 kdc1.example.com
_kpasswd._udp.EXAMPLE.COM       IN      SRV  0 0 749 kdc1.example.com
```

▼ 如何禁用票务授予票证的验证

此过程禁用用于检查本地 /etc/krb5/krb5.keytab 文件中存储的主机主体的 KDC 是否即是颁发票证授予票证的 KDC 的安全检查。该检查目的在于防止 DNS 欺骗攻击。但是，对于某些客户机配置，主机主体可能不可用，因此需要禁用此项检查以允许客户机正常工作。以下是需要禁用此项检查的配置：

- 动态分配客户机 IP 地址。例如，DHCP 客户机。
- 客户机未配置为托管任何服务，所以没有创建主机主体。
- 未在客户机上储存主机密钥。

1 成为超级用户。

2 更改 `krb5.conf` 文件。

如果 `verify_ap_req_nofail` 选项设置为 `false`，则不会启用 TGT 验证过程。有关该选项的更多信息，请参见 [krb5.conf\(4\)](#) 手册页。

```
client # cat /etc/krb5/krb5.conf
[libdefaults]
    default_realm = EXAMPLE.COM
    verify_ap_req_nofail = false
...
```

注 – 可以在 `krb5.conf` 文件的 `[libdefaults]` 或 `[realms]` 部分中输入 `verify_ap_req_nofail` 选项。如果此选项位于 `[libdefaults]` 部分，则该设置用于所有领域。如果该选项位于 `[realms]` 部分，则该设置仅应用于所定义的领域。

▼ 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 特权访问要求 Kerberos 验证的 NFS 文件系统。特别是，使用如下选项共享 NFS 文件系统时：`-o sec=krb5,root=client1.sun.com`。

● 启动 `kadmin`。

可以使用 GUI Kerberos 管理工具添加主体，如第 421 页中的“如何创建新的 Kerberos 主体”所述。为此，必须使用在配置主 KDC 服务器时创建的一个 `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 NFS 挂载的文件系统提供 `root` 等效访问权限。`root` 主体应有两个组成部分，第二个部分是 Kerberos 客户机系统的主机名，以避免创建领域范围的 `root` 主体。请注意，当主体实例为主机名时，无论名称服务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具有 root 访问权限。如果需要非交互式 root 访问权限（如以 root 身份运行 cron 作业），也必须执行该步骤。

```
kadmin: ktadd root/client.example.com
Entry for principal root/client.example.com with kvno 3, encryption type AES-256 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 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
```

▼ 如何在 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。可以使用正在使用的服务的 PAM 框架，通过在 `/etc/pam.conf` 中将 `pam_krb5_migrate` 模块堆叠在该服务的验证栈中实现迁移。

在此示例中，将配置 `gdm` 和 `other` PAM 服务名称使其使用自动迁移。将使用以下配置参数：

- 领域名称 = `EXAMPLE.COM`
- 主 KDC = `kdc1.example.com`
- 托管迁移服务的计算机 = `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 351 页中的“配置 Kerberos 客户机”。

- 1 成为超级用户。
- 2 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 服务器验证该服务器。

```
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
```

3 更改 PAM 配置文件。

a. 为 gdm 服务添加项。

```
# cat /etc/pam.conf
.
.
#
# gdm service
#
gdm      auth requisite      pam_authok_get.so.1
gdm      auth required      pam_dhkeys.so.1
gdm      auth required      pam_unix_cred.so.1
gdm      auth sufficient     pam_krb5.so.1
gdm      auth requisite      pam_unix_auth.so.1
gdm      auth optional       pam_krb5_migrate.so.1
```

b. 可选如果需要，可强制立即更改口令。

可以将新建 Kerberos 帐户的口令截止时间设置为当前时间（现在），以便强制立即更改 Kerberos 口令。要将截止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
.
.
gdm      auth optional       pam_krb5_migrate.so.1 expire_pw
```

c. 将 `pam_krb5` 模块添加到帐户栈。

此添加操作使 Kerberos 中的口令失效，从而阻止访问。

```
# cat /etc/pam.conf
.
.
#
# Default definition for Account management
# Used when service name is not explicitly mentioned for account management
#
other    account requisite    pam_roles.so.1
other    account required     pam_krb5.so.1
other    account required     pam_unix_account.so.1
```

d. 将 `pam_krb5` 模块添加到口令栈。

此添加操作使得口令在到期时得到更新。

```
# cat /etc/pam.conf
.
.
#
# Default definition for Password management
# Used when service name is not explicitly mentioned for password management
#
```

```
other password required pam_dhkeys.so.1
other password requisite pam_authtok_get.so.1
other password requisite pam_authtok_check.so.1
other password sufficient pam_krb5.so.1
other password required pam_authtok_store.so.1
```

4 在主 KDC 服务器上，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特权。务必将不应被迁移的用户在 `kadm5.acl` 文件中以 `U` 特权列出。这些项必须位于 `permit all` 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 服务器上，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 服务器上，向 `pam.conf` 文件添加项。

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
```

▼ 如何配置帐户锁定

● 启动 `kadmin`。

```
denver # /usr/sbin/kadmin -p kws/admin
Enter password: <Type kws/admin password>
kadmin:
```

a. 使用帐户锁定参数创建策略。

在下面的示例中，`add_policy` 子命令用于创建名为 `default` 的策略。在 300 秒内的三次验证失败将触发 900 秒的帐户锁定。

```
kadmin: add_policy -maxfailure 3 -failurecountinterval "300 seconds" \
-lockoutduration "900 seconds" default
```

b. 退出 `kadmin`。

```
kadmin: quit
```

示例 21-15 解锁锁定的主体

在下面的示例中，将解锁用户主体：

```
# kadmin
kadmin: add_policy -unlock principal
```

同步 KDC 与 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。如果任意两个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。

时钟相位差还决定应用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。

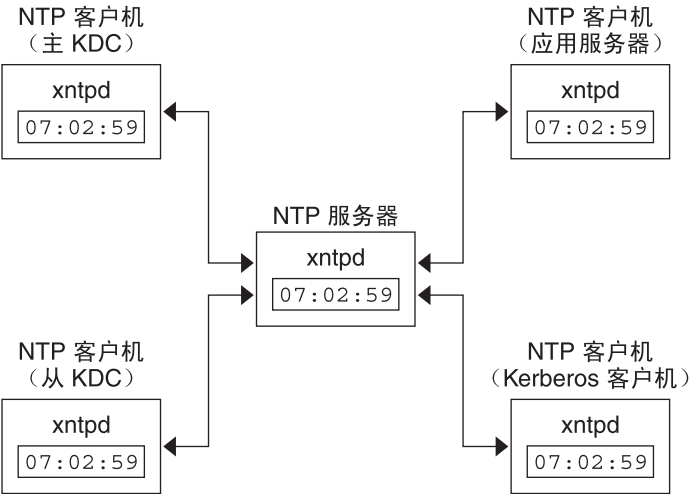
保持 KDC 与 Kerberos 客户机之间的时钟同步非常重要，因此应使用网络时间协议 (Network Time Protocol, NTP) 软件同步这些时钟。Oracle Solaris 软件中包括了由美国特拉华大学开发的 NTP 公共域软件。

注 – 另一种同步时钟的方法是使用 `rdate` 命令和 `cron` 作业，这是一种比使用 NTP 更简单的过程。不过，本节将着重介绍如何使用 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，您可以在网络环境中管理精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。以下给出了该服务器/客户机 NTP 实现的示例。

图 21-1 使用 NTP 同步时钟



确保 KDC 与 Kerberos 客户机保持时钟同步需要实现以下步骤：

1. 在网络上设置一个 NTP 服务器。此服务器可以是主 KDC 服务器以外的任何系统。要了解 NTP 服务器任务，请参见《[Oracle Solaris 管理：网络服务](#)》中的“[管理网络时间协议（任务）](#)”。
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《[Oracle Solaris 管理：网络服务](#)》中的“[管理网络时间协议（任务）](#)”。

交换主 KDC 服务器与从 KDC 服务器

使用本节中的过程可以使交换主 KDC 服务器与从 KDC 服务器的任务更轻松。仅当主 KDC 服务器由于某种原因出现故障，或者需要重新安装主 KDC 服务器（例如，由于安装了新硬件）时，才应将主 KDC 服务器与从 KDC 服务器进行交换。

▼ 如何配置可交换的从 KDC 服务器

在您希望成为主 KDC 服务器的从 KDC 服务器上执行此过程。该过程假定使用了增量传播。

- 1 在安装 KDC 过程中，使用主 KDC 服务器和可交换从 KDC 服务器的别名。
定义 KDC 的主机名时，请确保 DNS 中包含每个系统的别名。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 服务器安装。

在进行任何交换之前，该服务器应该充当领域中的任何其他从 KDC 服务器。有关说明，请参见第 337 页中的“如何手动配置从 KDC 服务器”。

3 移动主 KDC 服务器命令。

要禁止从该从 KDC 服务器运行主 KDC 服务器命令，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 服务器与从 KDC 服务器

在此过程中，要交换出来的主 KDC 服务器名为 `kdc1`。将成为新的主 KDC 服务器的从 KDC 服务器名为 `kdc4`。该过程假定使用了增量传播。

开始之前 此过程要求已将从 KDC 服务器设置为可交换的从 KDC 服务器。有关更多信息，请参见第 368 页中的“如何配置可交换的从 KDC 服务器”。

1 成为超级用户。

2 在新的主 KDC 服务器上，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。

以下示例中将第一个 `addprinc` 命令显示为两行，但实际上应在同一行中键入该命令。

```
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. 退出 `kadmin`。

```
kadmin: quit
```

3 在新的主 KDC 服务器上，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.uolog
```

- 4 在新的主 KDC 服务器上，确认更新是否完成。

```
kdc4 # /usr/sbin/kproplog -h
```

- 5 在新的主 KDC 服务器上，重新启动 KDC 服务。

```
kdc4 # svcadm enable -r network/security/krb5kdc
```

- 6 在新的主 KDC 服务器上，清除更新日志。

这些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.ulong
```

- 7 在旧的主 KDC 服务器上，中止 kadmind 和 krb5kdc 进程。

中止 kadmind 进程可防止对 KDC 数据库进行任何更改。

```
kdc1 # svcadm disable network/security/kadmind
kdc1 # svcadm disable network/security/krb5kdc
```

- 8 在旧的主 KDC 服务器上，指定请求传播的轮询时间。

注释掉 /etc/krb5/kdc.conf 中的 sunw_dbprop_master_ulogsize 项，并添加定义 sunw_dbprop_slave_poll 的项。该项将轮询时间设为两分钟。

```
kdc1 # cat /etc/krb5/kdc.conf
[kdcdefaults]
    kdc_ports = 88,750

[realms]
    EXAMPLE.COM= {
        profile = /etc/krb5/krb5.conf
        database_name = /var/krb5/principal
        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
        sunw_dbprop_slave_poll = 2m
    }
```

- 9 在旧的主 KDC 服务器上，移动主 KDC 服务器命令和 kadm5.acl 文件。

要禁止运行主 KDC 服务器命令，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
```

- 10 在 DNS 服务器上，更改主 KDC 服务器的别名。

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
```

- 11 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
```

- 12 在新的主 KDC 服务器上，移动主 KDC 服务器命令和从 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
```

- 13 在新的主 KDC 服务器上，创建 Kerberos 访问控制列表文件 (kadm5.acl)。

填充后，/etc/krb5/kadm5.acl 文件应包含所有获许管理 KDC 的主体名称。该文件还应列出请求增量传播的所有从 KDC 服务器。有关更多信息，请参见 [kadm5.acl\(4\)](#) 手册页。

```
kdc4 # cat /etc/krb5/kadm5.acl
kws/admin@EXAMPLE.COM *
kiprop/kdc1.example.com@EXAMPLE.COM p
```

- 14 在新的主 KDC 服务器上，在 kdc.conf 文件中指定更新日志大小。

注释掉 sunw_dbprop_slave_poll 项，并添加定义 sunw_dbprop_master_ulogsize 的项。该项将日志大小设置为 1000 个项。

```
kdc1 # cat /etc/krb5/kdc.conf
[kdcdefaults]
    kdc_ports = 88,750

[realms]
    EXAMPLE.COM= {
        profile = /etc/krb5/krb5.conf
        database_name = /var/krb5/principal
        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
        sunw_dbprop_master_ulogsize = 1000
    }
```

- 15 在新的主 KDC 服务器上，启动 kadmind 和 krb5kdc。

```
kdc4 # svcadm enable -r network/security/krb5kdc
kdc4 # svcadm enable -r network/security/kadmind
```

- 16 在旧的主 KDC 服务器上，添加 kiprop 服务主体。

通过将 kiprop 主体添加到 krb5.keytab 文件，允许 kpropd 守护进程对自身进行增量传播服务验证。

```
kdc1 # /usr/sbin/kadmin -p kws/admin
Authenticating as principal 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-256 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 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

```

- 17 在旧的主 KDC 服务器上，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

```

- 18 在旧的主 KDC 服务器上，启动 **kpropd** 和 **krb5kdc**。

```

kdc1 # svcadm enable -r network/security/krb5_prop
kdc1 # svcadm enable -r 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 325 页中的“如何手动配置主 KDC 服务器”中介绍了该数据库的初始设置步骤。

备份和传播 Kerberos 数据库

将 Kerberos 数据库从主 KDC 服务器传播到从 KDC 服务器是最重要的配置任务之一。如果传播发生频率不够高，主 KDC 服务器与从 KDC 服务器将无法保持同步。因此，如果主 KDC 服务器关闭，从 KDC 服务器将不能获取最新的数据库信息。此外，如果出于负载平衡目的将从 KDC 服务器配置为主 KDC 服务器，则将该从 KDC 服务器用作主 KDC 服务器的客户机将不能获取最新的信息。所以，必须确保传播发生频率足够高，或者基于更改 Kerberos 数据库的频率配置服务器以进行增量传播。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也较低。

配置主 KDC 服务器时，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，并将该数据库传播到从 KDC 服务器。不过，与其他任何文件一样，Kerberos 数据库可能会受损。如果在从 KDC 服务

器上发生数据受损，您可能不会注意到，因为下一次数据库自动传播会安装一个全新的副本。但是，如果在主 KDC 服务器上发生数据受损，则在下一次传播期间会将损坏的数据库传播到所有从 KDC 服务器。而且，损坏的备份会覆盖主 KDC 服务器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` 转储文件复制到其他位置，或者使用 `kdb5_util` 的 `dump` 命令创建另一个单独的备份副本。这样，即使数据库受损，也可以使用 `kdb5_util` 的 `load` 命令在主 KDC 服务器上恢复最新的备份。

另一个重要注意事项是：数据库转储文件包含主体密钥，因此需要防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。要防止未经授权的访问，请仅使用 `kprop` 命令传播数据库转储文件，该命令会对传送中的数据进行了加密。此外，`kprop` 命令仅将数据传播到从 KDC 服务器，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，而在下一次传播之前该数据库受损，则从 KDC 服务器将不包含这些更新。这些更新将会丢失。因此，如果在计划的定期传播之前向 Kerberos 数据库添加了重要的更新，应手动传播该数据库，以避免数据丢失。

kpropd.acl 文件

从 KDC 服务器上的 `kpropd.acl` 文件提供主机主体名称的列表（每个名称占一行），用于指定该 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 服务器传播到所有从 KDC 服务器，则每个从 KDC 服务器上的 `kpropd.acl` 文件仅需包含主 KDC 服务器的主机主体名称。

但是，本书中的 Kerberos 安装和后续配置步骤将引导您将同一个 `kpropd.acl` 文件添加到主 KDC 服务器和从 KDC 服务器。该文件包含所有 KDC 主机主体名称。通过此配置，在传播端 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，可以轻松地维护配置。

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 服务器上运行 `kprop_script` 命令，则会将该从 KDC 服务器的 Kerberos 数据库副本传播到其他 KDC。`kprop_script` 接受主机名列表作为参数，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 成为管理员，或者承担已指定给 Kerberos 服务器管理配置文件的角色或用户名。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。

- 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` 显示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。可以指定该文件的绝对路径。如果未指定文件，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 21-16 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。由于指定了 `-verbose` 选项，备份时会显示每个主体。

```
# 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 服务器上的超级用户。
- 2 在主 KDC 服务器上，停止 KDC 守护进程。

```
kdc1 # svcadm disable network/security/krb5kdc
kdc1 # svcadm disable network/security/kadmin
```

3 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` 显示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盖现有数据库。

`filename` 定义用于恢复数据库的文件。可以指定该文件的绝对路径。

4 启动 KDC 守护进程。

```
kdc1 # svcadm enable -r network/security/krb5kdc
kdc1 # svcadm enable -r network/security/kadmin
```

示例 21-17 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
```

▼ 如何在服务升级后转换 Kerberos 数据库

如果您的 KDC 数据库是在运行 Solaris 8 或 Solaris 9 发行版的服务器上创建的，转换该数据库可允许您利用改进的数据库格式。

开始之前 确保数据库使用的是旧格式。

1 在主 KDC 服务器上，停止 KDC 守护进程。

```
kdc1 # svcadm disable network/security/krb5kdc
kdc1 # svcadm disable network/security/kadmin
```

2 创建一个目录用来存储数据库的临时副本。

```
kdc1 # mkdir /var/krb5/tmp
kdc1 # chmod 700 /var/krb5/tmp
```

3 转储 KDC 数据库。

```
kdc1 # kdb5_util dump /var/krb5/tmp/prdb.txt
```

4 保存当前数据库文件的副本。

```
kdc1 # cd /var/krb5
kdc1 # mv princ* tmp/
```

5 装入数据库。

```
kdc1 # kdb5_util load /var/krb5/tmp/prdb.txt
```

6 启动 KDC 守护进程。

```
kdc1 # svcadm enable -r network/security/krb5kdc
kdc1 # svcadm enable -r network/security/kadmin
```

▼ 如何重新配置主 KDC 服务器以使用增量传播

此过程中的步骤可用于重新配置现有的主 KDC 服务器，以使用增量传播。此过程中使用以下配置参数：

- 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 成为超级用户。

2 向 kdc.conf 中添加项。

需要启用增量传播，并选择主 KDC 服务器在日志中记录的更新的数目。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
        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_uologsize = 1000
    }
```

3 创建 kiprop 主体。

kiprop 主体用于验证主 KDC 服务器以及授权来自主 KDC 服务器的更新。

```
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 "kprop/kdc2.example.com@EXAMPLE.COM" created.
kadmin:
```

4 在主 KDC 服务器上，向 `kadm5.acl` 中添加 `kprop` 项。

通过此项，主 KDC 服务器可以接收来自 `kdc2` 服务器的增量传播请求。

```
kdc1 # cat /etc/krb5/kadm5.acl
*/admin@EXAMPLE.COM *
kprop/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 服务器传播其 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。

有关完整说明，请参见第 377 页中的“如何重新配置从 KDC 服务器以使用增量传播”。

▼ 如何重新配置从 KDC 服务器以使用增量传播

1 成为超级用户。

2 向 `kdc.conf` 中添加项。

第一个新项将启用增量传播。第二个新项将轮询时间设为两分钟。

```
kdc2 # cat /etc/krb5/kdc.conf
[kdcdefaults]
    kdc_ports = 88,750

[realms]
    EXAMPLE.COM= {
        profile = /etc/krb5/krb5.conf
        database_name = /var/krb5/principal
        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
    }

```

3 将 kiproop 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 kiproop/kdc2.example.com
Entry for principal kiproop/kdc2.example.com with kvno 3, encryption type AES-256 CTS mode
with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal kiproop/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 kiproop/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 kiproop/kdc2.example.com with kvno 3, encryption type ArcFour
with HMAC/md5 added to keytab WRFILE:/etc/krb5/krb5.keytab.
Entry for principal kiproop/kdc2.example.com with kvno 3, encryption type DES cbc mode
with RSA-MD5 added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin: quit

```

4 重新启动 kpropd。

```
kdc2 # svcadm restart network/security/krb5_prop
```

▼ 如何配置从 KDC 服务器以使用完全传播

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，只有当主 KDC 服务器运行 Solaris 9 发行版或更早发行版时才使用此过程。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 服务器以允许进行传播。

在此过程中，将配置名为 kdc3 的从 KDC 服务器。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://download.oracle.com/docs/cd/E23824_01/html/821-1456/aadmin-23.html

注 – 调整 URL 使其指向本部分，如第 320 页中的“图形化 Kerberos 管理工具中的联机帮助 URL”中所述。

开始之前 必须配置主 KDC 服务器。要了解交换从 KDC 服务器的具体说明，请参见第 368 页中的“交换主 KDC 服务器与从 KDC 服务器”。

1 成为主 KDC 服务器上的超级用户。

2 在主 KDC 服务器上，启动 `kadmin`。

必须使用在配置主 KDC 服务器时创建的一个 `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 服务器上，向数据库中添加从主机主体（如果尚未完成）。

要使从 KDC 服务器正常工作，它必须有主机主体。请注意，当主体实例为主机名时，无论名称服务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 服务器上，编辑 Kerberos 配置文件 (`krb5.conf`)。

需要为每个从 KDC 服务器添加一个项。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 服务器上，将主 KDC 服务器和每个从 KDC 服务器的项添加到 `kpropd.acl` 文件中。

有关此文件的完整说明，请参见 [kpropd\(1M\)](#) 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 在所有的从 KDC 服务器上，从主 KDC 服务器复制 KDC 管理文件。

需要在所有的从 KDC 服务器上执行该步骤，因为主 KDC 服务器已经更新了每个 KDC 服务器需要的信息。可以使用 ftp 或类似的传输机制从主 KDC 服务器获取以下文件的副本：

- /etc/krb5/krb5.conf
- /etc/krb5/kdc.conf
- /etc/krb5/kpropd.acl

6 在所有的从 KDC 服务器上，确保未填充 Kerberos 访问控制列表文件 kadm5.acl。

未修改的 kadm5.acl 文件类似如下示例：

```
kdc2 # cat /etc/krb5/kadm5.acl
*/admin@__default_realm__ *
```

如果该文件中包含 kprop 项，请将其删除。

7 在新的从 KDC 服务器上，启动 kadmin 命令。

必须使用在配置主 KDC 服务器时创建的一个 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 服务器的 host 主体添加到从 KDC 服务器的密钥表文件中。

该项可使 kprop 及其他基于 Kerberos 的应用程序正常工作。请注意，当主体实例为主机名时，无论名称服务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-256 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 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 服务器上，将从 KDC 服务器名称添加到 `cron` 作业中，该作业通过运行 `crontab -e` 自动运行备份。

在 `kprop_script` 行的末尾添加每个从 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。此项将在每天的凌晨 3:10 启动备份过程。

- 9 在新的从 KDC 服务器上，启动 Kerberos 传播守护进程。

```
kdc3 # svcadm enable network/security/krb5_prop
```

- 10 在主 KDC 服务器上，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，则不必完成其他备份。有关进一步的说明，请参见第 382 页中的“如何手动将 Kerberos 数据库传播到从 KDC 服务器”。

```
kdc1 # /usr/lib/krb5/kprop_script kdc3.example.com
Database propagation to kdc3.example.com: SUCCEEDED
```

- 11 在新的从 KDC 服务器上，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 可选在新的从 KDC 服务器上，使用 NTP 或其他时钟同步机制同步主 KDC 服务器时钟。

安装和使用网络时间协议 (Network Time Protocol, NTP) 并非必须的。但是，要成功进行验证，每个时钟必须遵守 `krb5.conf` 文件的 `libdefaults` 部分中定义的缺省时间。有关 NTP 的信息，请参见第 367 页中的“同步 KDC 与 Kerberos 客户机的时钟”。

- 13 在新的从 KDC 服务器上，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 服务器上的信息。

- 1 成为超级用户。
- 2 在主 KDC 服务器上，运行 `kproplog` 命令。

```
kdc1 # /usr/sbin/kproplog -h
```
- 3 在从 KDC 服务器上，运行 `kproplog` 命令。

```
kdc2 # /usr/sbin/kproplog -h
```

4 检查最后一个序列号和最后一个时间戳的值是否匹配。

示例 21-18 检验 KDC 服务器是否同步

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.ulong)
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.ulong)
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戳的值相同，表示从 KDC 服务器与主 KDC 服务器同步。

在从 KDC 服务器的输出中，可以注意到从 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戳不是相关信息，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 服务器

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 服务器与主 KDC 服务器同步，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 成为管理员，或者承担已指定给 Kerberos 服务器管理配置文件的角色或用户名。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。
- 2 成为主 KDC 服务器上的超级用户。
- 3 可选使用 `kdb5_util` 命令备份数据库。
`# /usr/sbin/kdb5_util dump /var/krb5/slave_datatrans`
- 4 使用 `kprop` 命令将数据库传播到从 KDC 服务器。
`# /usr/lib/krb5/kprop -f /var/krb5/slave_datatrans slave-KDC`

示例 21-19 使用 `kprop_script` 手动将 Kerberos 数据库传播到从 KDC 服务器

如果要备份数据库，并在定期的 `cron` 作业之外将该数据库传播到从 KDC 服务器，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

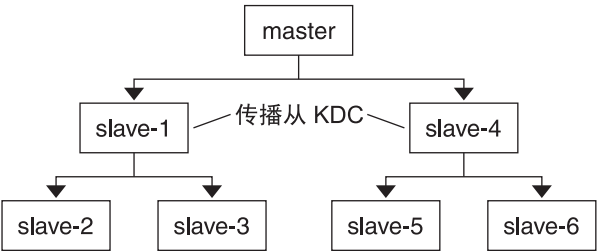
大多数情况下，会以独占的方式使用主 KDC 服务器将其 Kerberos 数据库传播到从 KDC 服务器。但是，如果站点上有很多从 KDC 服务器，可以考虑分担传播过程的负荷，称为**并行传播**。

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 服务器可以与主 KDC 服务器分担传播工作负荷。通过这种分担，可以更快地完成传播并减轻主 KDC 服务器的工作负荷。

例如，假设站点上有一个主 KDC 服务器和六个从 KDC 服务器（如图 21-2 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 服务器将数据库传播到 `slave-1` 和 `slave-4`。然后，这些从 KDC 服务器又可将该数据库传播到自己组中的其他从 KDC 服务器。

图 21-2 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概要列表。这些步骤包括：

1. 在主 KDC 服务器上，更改其 cron 作业中的 `kprop_script` 项，以仅包括将执行后续传播的从 KDC 服务器（**从 KDC 传播服务器**）的参数。
2. 在每个从 KDC 传播服务器上，将 `kprop_script` 项添加到其 cron 作业中，其中必须包括要传播的从 KDC 服务器的参数。要成功实现并行传播，应设置 cron 作业，使其在将新 Kerberos 数据库传播到从 KDC 传播服务器本身之后再运行。

注 - 向从 KDC 传播服务器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 服务器上，设置适当的传播权限。通过将传播端 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 21-20 设置并行传播

以图 21-2 为例，主 KDC 服务器的 `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` 项类似于以下示例：

```
0 4 * * * /usr/lib/krb5/kprop_script slave-2.example.com slave-3.example.com
```

请注意，从 KDC 服务器上的传播在主 KDC 服务器将数据库传播到它一小时后开始。

从 KDC 传播服务器上的 `kpropd.acl` 文件将包含以下项：

```
host/master.example.com@EXAMPLE.COM
```

要从 slave-1 传播到的从 KDC 服务器上的 `kpropd.acl` 文件将包含以下项：

示例 21-20 设置并行传播 (续)

```
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，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 数据库后，才需要删除存储文件。由于存储文件不会自动随数据库一起删除，所以您必须手动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 上的超级用户。
- 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`。

注 – 如果需要重新创建存储文件，可以使用 `kdb5_util` 命令的 `-f` 选项。

▼ 如何使用新的主密钥

- 1 成为管理员，或者承担已指定给 Kerberos 服务器管理配置文件的角色或用户名。有关更多信息，请参见第 150 页中的“如何获取管理权限”。
- 2 创建一个新的密钥。

此命令将添加一个新的随机生成的主密钥。`-s` 选项要求将新的主密钥存储在缺省的密钥表中。

```
# kdb5_util add_mkey -s
```

```
Creating new master key for master key principal 'K/M@EXAMPLE.COM'
You will be prompted for a new database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key:      <Type the password>
Re-enter KDC database master key to verify:  <Type it again>
```

3 验证新的主密钥是否存在。

```
# kdb5_util list_mkeys
Master keys for Principal: K/M@EXAMPLE.COM
KNVO: 2, Enctype: AES-128 CTS mode with 96-bit SHA-1 HMAC, No activate time set
KNVO: 1, Enctype: DES cbc mode with RSA-MD5, Active on: Wed Dec 31 18:00:00 CST 2001 *
```

此输出中的星号标识当前处于活动状态的主密钥。

4 设置新创建的主密钥变为活动状态的时间。

```
# date
Fri Jul 1 17:57:00 CDT 2011
# kdb5_util use_mkey 2 'now+2days'
# kdb5_util list_mkeys
Master keys for Principal: K/M@EXAMPLE.COM
KNVO: 2, Enctype: AES-128 CTS mode with 96-bit SHA-1 HMAC, Active on: Sun Jul 03 17:57:15 CDT 2011
KNVO: 1, Enctype: DES cbc mode with RSA-MD5, Active on: Wed Dec 31 18:00:00 CST 2001 *
```

在下面的示例中，日期设置为未来的两天，这样新的主密钥便有时间传播到所有 KDC。调整日期，使其适合您的环境。

5 可选创建新的主体后，确保正在使用新的主密钥。

```
# kadmin.local -q 'getprinc jimf' | egrep 'Principal|MKey'
Authenticating as principal root/admin@EXAMPLE.COM with password.
Principal: jimf@EXAMPLE.COM
MKey: vno 2
```

在下面的示例中，MKey: vno 2 指示主体的密钥受新创建的主密钥 (2) 的保护。

6 使用新的主密钥重新加密用户主体密钥。

如果将模式参数添加到命令的末尾，将会更新与此模式相匹配的主体。向此命令语法添加 -n 选项，以标识将要更新的主体。

```
# kdb5_util update_princ_encryption -f -v
Principals whose keys WOULD BE re-encrypted to master key vno 2:
updating: host/kdc1.example.com@EXAMPLE.COM
skipping: jimf@EXAMPLE.COM
updating: kadmin/changepw@EXAMPLE.COM
updating: kadmin/history@EXAMPLE.COM
updating: kdc/admin@EXAMPLE.COM
updating: host/kdc2.example.com@EXAMPLE.COM
6 principals processed: 5 updated, 1 already current
```

7 清除旧的主密钥。

当某个主密钥不再用于保护任何主体密钥时，可将其从主密钥主体中清除。此命令不会清除那些仍被主体使用的密钥。向此命令添加 -n 选项，以确保将清除正确的主密钥。

```
# kdb5_util purge_mkeys -f -v
Purging the follwing master key(s) from K/M@EXAMPLE.COM:
KNVO: 1
1 key(s) purged.
```

8 验证是否已清除旧的主密钥。

```
# kdb5_util list_mkeys
Master keys for Principal: K/M@EXAMPLE.COM
KNVO: 2, Enctype: AES-128 CTS mode with 96-bit SHA-1 HMAC, Active on: Sun Jul 03 17:57:15 CDT 2011 *
```

9 更新存储文件。

```
# kdb5_util stash
Using existing stashed keys to update stash file.
```

10 验证存储文件是否已更新。

```
# klist -kt /var/krb5/.k5.EXAMPLE.COM
Keytab name: FILE:.k5.EXAMPLE.COM
KVNO Timestamp Principal
-----
2 05/07/2011 15:08 K/M@EXAMPLE.COM
```

在 LDAP 目录服务器上管理 KDC

大多数使用 LDAP 目录服务器管理 KDC 的任务与使用 DB2 服务器相同。不过有一些特定于使用 LDAP 的新任务。

表 21-3 配置 KDC 服务器以使用 LDAP（任务列表）

任务	说明	参考
配置主 KDC。	通过使用手动过程并将 KDC 配置为使用 LDAP，为领域配置并构建主 KDC 服务器和数据库。	第 329 页中的“如何配置 KDC 以使用 LDAP 数据服务器”
混合使用 Kerberos 主体属性与非 Kerberos 对象类类型。	允许将信息存储在 Kerberos 记录中，以与其他 LDAP 数据库共享。	第 387 页中的“如何在非 Kerberos 对象类类型中混合 Kerberos 主体属性”
销毁领域。	删除与领域关联的所有数据。	第 388 页中的“如何在 LDAP 目录服务器上销毁领域”

▼ 如何在非 Kerberos 对象类类型中混合 Kerberos 主体属性

此过程允许 Kerberos 主体属性与非 Kerberos 对象类类型关联。在该过程中，krbprincipalaux、krbTicketPolicyAux 和 krbPrincipalName 属性都与 people 对象类关联。

此过程中使用以下配置参数：

- 目录服务器 = dsserver.example.com
- 用户主体 = willf@EXAMPLE.COM

1 成为超级用户。

2 准备 **people** 对象类中的每一项。

对每一项重复此步骤。

```
cat << EOF | ldapmodify -h dsserver.example.com -D "cn=directory manager"
dn: uid=willf,ou=people,dc=example,dc=com
changetype: modify
objectClass: krbprincipalaux
objectClass: krbTicketPolicyAux
krbPrincipalName: willf@EXAMPLE.COM
EOF
```

3 将一个子树属性添加到领域容器。

此步骤允许在 `ou=people,dc=example,dc=com` 容器和缺省 `EXAMPLE.COM` 容器中搜索主体项。

```
# kdb5_ldap_util -D "cn=directory manager" modify \
    -subtrees 'ou=people,dc=example,dc=com' -r EXAMPLE.COM
```

4 可选如果 KDC 记录存储在 DB2 中，迁移 DB2 项。

a. 转储 DB2 项。

```
# kdb5_util dump > dumpfile
```

b. 将数据库装入 LDAP 服务器。

```
# kdb5_util load -update dumpfile
```

5 可选将主体属性添加到 KDC。

```
# kadmin.local -q 'addprinc willf'
```

▼ 如何在 LDAP 目录服务器上销毁领域

此过程可用于已经将其他 LDAP 目录服务器配置为处理领域的情况。

1 成为超级用户。

2 销毁领域。

```
# kdb5_ldap_util -D "cn=directory manager" destroy
```


增强 Kerberos 服务器的安全性

执行以下步骤可以增强 Kerberos 应用服务器和 KDC 服务器的安全性。

表 21-4 增强 Kerberos 服务器的安全性（任务列表）

任务	说明	参考
启用使用 Kerberos 验证的访问。	限制对服务器的网络访问，以仅允许 Kerberos 验证。	第 389 页中的“如何仅启用基于 Kerberos 的应用程序”
限制对 KDC 服务器的访问。	增强 KDC 服务器及其数据的安全性。	第 389 页中的“如何限制对 KDC 服务器的访问”
通过使用字典文件增强口令的安全性。	依据字典检查新口令，以增强其安全性。	第 390 页中的“如何使用字典文件提高口令的安全性”

▼ 如何仅启用基于 Kerberos 的应用程序

此过程限制对运行 telnet、ftp、rcp、rsh 和 rlogin 的服务器的网络访问，以仅使用通过 Kerberos 验证的事务。

- 1 成为超级用户。
- 2 更改 telnet 服务的 exec 属性。
将 -a user 选项添加到 telnet 的 exec 属性，以将访问限于可以提供有效验证信息的用户。

```
# inetadm -m svc:/network/telnet:default exec="/usr/sbin/in.telnetd -a user"
```
- 3 可选如果尚未配置，请更改 telnet 服务的 exec 属性。
将 -a 选项添加到 ftp 的 exec 属性，以仅允许通过 Kerberos 验证的连接。

```
# inetadm -m svc:/network/ftp:default exec="/usr/sbin/in.ftpd -a"
```
- 4 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 数据库的副本。限制对这些服务器的访问以保证数据库安全对于 Kerberos 安装的整体安全性非常重要。

- 1 成为超级用户。

2 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。根据配置的不同，可能已禁用其中某些服务。使用 `svcs` 命令检查服务状态。大多数情况下，如果 KDC 是从服务器，仅需要运行 `krb5kdc` 和 `krdb5_kprop` 服务；如果 KDC 是主服务器，则仅需要运行 `kadmin` 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
```

3 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场所。用户不应能够以任何方式访问此服务器。

4 将 KDC 数据库备份存储在本地磁盘或从 KDC 服务器上。

仅在可以安全存储磁带时才应创建 KDC 的磁带备份。此做法同样适用于密钥表文件的备份。最好在不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 服务器上。

▼ 如何使用字典文件提高口令的安全性

Kerberos 服务可以使用字典文件来防止在创建新凭证时将字典中的词用作口令。防止将字典术语用作口令可以让其他人更难猜到口令。缺省情况下使用 `/var/krb5/kadm5.dict` 文件，但它是空的。

1 成为主 KDC 服务器上的超级用户。

2 编辑 KDC 配置文件 (`kdc.conf`)。

您需要添加一个行来指示该服务使用字典文件。在此示例中，使用包括在 `spell` 实用程序中的字典。有关配置文件的完整说明，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
        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
dict_file = /usr/share/lib/dict/words
}
```

3 重新启动 Kerberos 守护进程。

```
kdc1 # svcadm restart -r network/security/krb5kdc
kdc1 # svcadm restart -r network/security/kadmin
```


Kerberos 错误消息和故障排除

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故障排除提示。以下是本章中错误消息和故障排除信息的列表。

- 第 393 页中的“SEAM Tool 错误消息”
- 第 394 页中的“常见的 Kerberos 错误消息 (A-M)”
- 第 403 页中的“常见的 Kerberos 错误消息 (N-Z)”
- 第 407 页中的“krb5.conf 文件的格式存在问题”
- 第 407 页中的“传播 Kerberos 数据库时出现问题”
- 第 407 页中的“挂载基于 Kerberos 的 NFS 文件系统时出现问题”
- 第 408 页中的“以 root 用户身份进行验证时出现问题”
- 第 408 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 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 (kadmind5.acl) 中没有列出特权 (l)。因此，无法查看主体列表或策略列表。

解决方法: 必须在 "Name" (名称) 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特权的主体登录。

```
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 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: 确保调用 rlogind 时使用了 -k 选项。

```
Another authentication mechanism must be used to access this host
```

原因: 无法完成验证。

解决方法: 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: 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, 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: 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:确保提供的值与 [kinit\(1\)](#) 手册页中“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:确保提供的值与 [kinit\(1\)](#) 手册页中“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:确保至少可以访问一个 KDC（主从皆可），或 `krb5kdc` 守护进程正在 KDC 上运行。有关已配置的 KDC (`kdc = kdc-name`) 的列表，请查看 `/etc/krb5/krb5.conf` 文件。

Cannot determine realm for host: host is '*hostname*'

原因:Kerberos 无法确定主机的领域名称。

解决方法: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 a kadmin KDC entry in `krb5.conf(4)` or DNS Service Location records for realm '*realmname*'

Cannot find a kpassword KDC entry in `krb5.conf(4)` or DNS Service Location records for realm '*realmname*'

Cannot find a master KDC entry in `krb5.conf(4)` or DNS Service Location records for realm '*realmname*'

Cannot find any KDC entries in `krb5.conf(4)` or DNS Service Location records for realm '*realmname*'

原因:`krb5.conf` 文件或 DNS 服务器记录配置不正确。

解决方法:确保 Kerberos 配置文件 (`/etc/krb5/krb5.conf`) 或 KDC 的 DNS 服务器记录已配置正确。

Cannot find address for '*hostname*': '*error-string*'

原因:在 DNS 记录中找不到给定主机名的地址。

解决方法:修复 DNS 中的主机记录，或更正 DNS 查找进程中的错误。

Cannot find KDC for requested realm

原因:在请求的领域中找到 KDC。

解决方法:确保 Kerberos 配置文件 (`krb5.conf`) 在 `realm` 部分指定了 KDC。

cannot initialize realm *realm-name*

原因:KDC 可能没有存储文件。

解决方法: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 无法确定领域的任何 KDC。

解决方法:确保 Kerberos 配置文件 (`krb5.conf`) 在 `realm` 部分指定了 KDC。

Cannot resolve network address for KDCs '*hostname*' discovered via DNS Service Location records for realm '*realm-name*'

Cannot resolve network address for KDCs '*hostname*' specified in `krb5.conf(4)` for realm '*realm-name*'

原因:`krb5.conf` 文件或 DNS 服务器记录配置不正确。

解决方法:确保 Kerberos 配置文件 (`/etc/krb5/krb5.conf`) 或 KDC 的 DNS 服务器记录已正确配置。

Cannot reuse password

原因:指定的口令之前已被该主体使用。

解决方法:请选择一个以前未曾选用的口令,至少不应使用 KDC 数据库中为每个主体保存的那些口令。该策略由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: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:确保 `krb5.conf` 文件位于正确的位置,并具有合适的权限。该文件应可由 `root` 写入,并可由其他任何用户读取。

Client '*principal*' not found in Kerberos database

原因:Kerberos 数据库中缺少主体。

解决方法:将客户机主体添加到 Kerberos 数据库。

Client '*principal*' pre-authentication failed

原因:对主体的验证失败。

解决方法:确保用户使用的是正确的口令。

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: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: '*client-principal*'
requesting ticket '*service-principal*'

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:确保与之通信的服务器与客户机处于同一领域, 或者确保领域配置正确。

Client or server has a null key

原因:主体拥有空密钥。

解决方法:使用 kadmin 的 cpw 命令修改主体, 使其拥有非空密钥。

Clock skew too great: '*client*' requesting ticket '*service-principal*' from KDC
'*KDC-hostname*' (*KDC-time*).Skew is *value*

Clock skew too great: '*client*' AP request with ticket for '*service-principal*'.Skew is
value (allowable *value*)

原因:客户机上报告的时间与 KDC 服务器或应用服务器上报告的时间之间存在的差异过大。

解决方法:配置网络时间协议 (Network Time Protocol, NTP) 以保持时钟同步。有关更多信息, 请参见第 367 页中的“同步 KDC 与 Kerberos 客户机的时钟”。

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:确保为主 KDC 指定了正确的主机名。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:请确保您拥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 命令确认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: 请检验以下两种情况:

- 确保您的凭证有效。使用 `kdestroy` 销毁票证, 然后使用 `kinit` 创建新的票证。
- 确保目标主机的密钥表文件的服务密钥版本正确。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, 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。

Decrypt integrity check failed for client 'principal' and server 'hostname'

原因: 您的票证可能无效。

解决方法: 确保您的凭证有效。使用 `kdestroy` 命令销毁票证, 然后使用 `kinit` 命令创建新票证。

Encryption could not be enabled. Goodbye.

原因: 无法与服务器协商加密。

解决方法: 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 find realm for *principal* in keytab

原因: *principal* 中包含的领域名称与密钥表文件中存储的主体领域名称不匹配。

解决方法: 确保主体使用的是正确的领域。

failed to obtain credentials cache

原因: 在 `kadmin` 初始化期间, `kadmin` 尝试获取 `admin` 主体的凭证时失败。

解决方法: 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: 确认您没有在 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: 检查 `/var/krb5/kdc.log` 文件, 找出发生此错误时记录的更具体的错误消息。

Hostname cannot be canonicalized for 'hostname': 'error-string'

原因: Kerberos 客户机找不到服务器的全限定主机名。

解决方法: 确保在 DNS 中定义了服务器主机名, 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: 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: 确保 krb5.conf 文件中的所有关系后面都跟有 "=" 符号和值。另外, 请确认每个子段中的括号都是成对出现的。

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: 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: 确保网络地址正确。使用 kdestroy 销毁票证, 然后使用 kinit 创建新的票证。

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: 尝试使用该服务之前, 请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 Kerberos 内部错误。

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:为主体指定的口令包含的口令类数目不足, 达不到主体策略强制要求的数量。

解决方法: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:释放内存并再次尝试运行 `kadmin`。

kadmin: Bad encryption type while changing host/*FQDN*'s key

原因:Solaris 10 8/07 发行版之后的基本版本中包括更多的缺省加密类型。客户机请求的加密类型可能不受运行该软件旧版本的 KDC 支持。

解决方法:有几种解决方案可以解决这一问题。最简单的一种首先列出:

1. 向 KDC 服务器添加 `SUNWcry` 和 `SUNWcryr` 软件包。这可以增加 KDC 支持的加密类型数量。
2. 在客户机上的 `krb5.conf` 中设置 `permitted_encetypes`, 使其不包括 `aes256` 加密类型。将需要对每台新的客户机执行此步骤。

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:确定是请求了 KDC 不允许的选项, 还是票证类型不可用。

KDC policy rejects request

原因:KDC 策略不允许该请求。例如, 对 KDC 的请求中不含 IP 地址。或者, 请求了转发, KDC 却不允许。

解决方法:确保使用的 `kinit` 具有正确的选项。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。可以使用 `kadmin` 修改策略或主体。

KDC reply did not match expectation: KDC not found. Probably got an unexpected realm referral

原因:KDC 回复中未包含所需的主体名称, 或者响应中的其他值不正确。

解决方法:确保您与之通信的 KDC 符合 RFC4120, 发送的请求是 Kerberos V5 请求, 并且该 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:检查提供的高速缓存位置是否正确。如有必要, 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: 检查提供的高速缓存位置是否正确。如有必要, 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: 检查提供的高速缓存位置是否正确。如有必要, 使用 `kinit` 删除 TGT 并获取新的 TGT。

Kerberos authentication failed

原因: Kerberos 口令错误, 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: 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, 确保具有有效凭证。

Key table entry not found

原因: 网络应用服务器的密钥表文件中不存在服务主体的对应项。

解决方法: 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 table file '*filename*' not found

原因: 指定的密钥表文件不存在。

解决方法: 创建该密钥表文件。

Key version *number* is not available for principal *principal*

原因: 密钥的密钥版本与应用服务器中的密钥版本不匹配。

解决方法: 使用 `klist -k` 命令检查应用服务器中的密钥版本。

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: 如果更改了服务密钥 (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: 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 模块缺失，或不是有效的可执行二进制文件。

解决方法: 确保 Kerberos PAM 模块在 `/usr/lib/security` 目录下，并且是有效的可执行二进制文件。另外，请确保 `/etc/pam.conf` 文件包含指向 `pam_krb5.so.1` 的正确路径。

`Looping detected getting initial creds: 'client-principal' requesting ticket 'service-principal'. Max loops is value. Make sure a KDC is available.`

原因: Kerberos 多次尝试获取初始票证，但均失败。

解决方法: 确保至少有一个 KDC 正在响应验证请求。

`Master key does not match database`

原因: 装入的数据库转储不是基于包含主密钥的数据库创建的。主密钥位于 `/var/krb5/.k5.REALM` 中。

解决方法: 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您的请求需要的凭证。

解决方法: 使用 `kdestroy` 销毁票证，然后使用 `kinit` 创建新的票证。

`Message out of order`

原因: 使用顺序保密机制发送的消息到达时发生顺序混乱。某些消息可能已在传输过程中丢失。

解决方法: 应该重新初始化 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: 确保消息在网络中正确发送。由于此消息还可能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 找不到凭证高速缓存 (/tmp/krb5cc_*uid*)。

解决方法: 确保凭证文件存在且可读。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 ('*filename*') matches desired name *principal*

原因: 尝试验证服务器时发生错误。

解决方法: 确保主机主体或服务主体位于服务器的密钥表文件中。

Operation requires "*privilege*" privilege

原因: 所使用的 admin 主体未在 kadm5.acl 文件中配置相应的特权。

解决方法: 使用具有相应特权的主体。或者, 通过修改 kadm5.acl 文件来配置所使用的主体, 使其具有相应特权。通常, 名称中包含 /admin 的主体都具有相应的特权。

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: 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: 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，重放高速缓存文件名为 `/var/krb5/rcache/rc_service_name_uid`。对于 root 用户，重放高速缓存文件名为 `/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: 确保应用程序使用的是 Kerberos V5 协议。

Request is a replay

原因: 请求已经发送到该服务器，并进行了处理。票证可能已被盗用，其他用户正在尝试重用这些票证。

解决方法: 等待几分钟，然后重新发送请求。

Requested principal and ticket don't match: Requested principal is '*service-principal*' and TGT principal is '*TGT-principal*'

原因: 您正连接的服务主体和您拥有的服务票证不匹配。

解决方法: 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: 确保应用程序使用的是 Kerberos V5 协议。

Service key *service-principal* not available

原因: 应用服务器的密钥表文件中不存在指定的服务主体。

解决方法: 确保服务主体与应用服务器的密钥表文件中的服务主体匹配，或包含在密钥表文件中。

**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: 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启用验证。

Server refused to negotiate encryption. Good bye.

原因: 无法与服务器协商加密。

解决方法: 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了验证。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。

Server service-principal not found in Kerberos database

原因: 服务主体不正确，或主体数据库中缺少服务主体。

解决方法: 确保服务主体正确，并且位于数据库中。

Target name principal '*principal*' does not match service-principal

原因: 所使用的服务主体与应用服务器所使用的服务主体不匹配。

解决方法: 在应用服务器中，确保服务主体包含在密钥表文件中。对于客户机，确保使用的是正确的服务主体。

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。原因可能是所发送票证使用的是主体的 FQDN 名称，而服务需要非 FQDN 名称，或者所发送票证使用的是主体的非 FQDN 名称，而服务需要 FQDN 名称。

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: 您的票证已到期。

解决方法: 使用 `kdestroy` 销毁票证，然后使用 `kinit` 创建新的票证。

Ticket is ineligible for postdating

原因: 主体不允许其票证在以后生效。

解决方法: 使用 `kadmin` 修改主体，以允许以后生效。

Ticket not yet valid: '*client-principal*' requesting ticket '*service-principal*' from '*kdc-hostname*' (*time*). TGT start time is *time*.

原因: 以后生效的票证尚未生效。

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: 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: 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, 确保具有有效凭证。

Unknown encryption type: *name*

原因: 凭证中包含的加密类型无法使用。

解决方法: 通过 `klist -e` 命令确定客户机所使用的加密类型。确保应用服务器至少支持一种加密类型。

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: 确保服务的主体与票证中的主体匹配。

Kerberos 故障排除

本节介绍了有关 Kerberos 软件的故障排除信息。

▼ 如何识别密钥版本号相关的问题

有时, KDC 使用的密钥版本号 (Key Version Number, KVNO) 与 `/etc/krb5/krb5.keytab` 中存储的系统托管服务的服务主体密钥不匹配。如果 KDC 中创建了一组新密钥, 但没有在密钥表文件中更新这些新密钥, KVNO 可能会不同步。可以通过以下过程诊断此问题。

1 列出 **keytab** 条目。

请注意, 每个主体的 KVNO 都包含在列表中。

```
# klist -k
Keytab name: FILE:/etc/krb5/krb5.keytab
KVNO Principal
-----
2 host/denver.example.com@EXAMPLE.COM
2 host/denver.example.com@EXAMPLE.COM
2 host/denver.example.com@EXAMPLE.COM
2 nfs/denver.example.com@EXAMPLE.COM
2 nfs/denver.example.com@EXAMPLE.COM
2 nfs/denver.example.com@EXAMPLE.COM
2 nfs/denver.example.com@EXAMPLE.COM
```

2 使用 host 密钥获取初始凭证。

```
# kinit -k
```

3 确定 KDC 所使用的 KVNO。

```
# kvno nfs/denver.example.com
nfs/denver.example.com@EXAMPLE.COM: kvno = 3
```

请注意，此处列出的 KVNO 是 3 而不是 2。

krb5.conf 文件的格式存在问题

如果 krb5.conf 文件的格式不正确，则以下错误消息可能会显示在终端窗口中或记录在日志文件中：

```
Improper format of Kerberos configuration file while initializing krb5 library
```

如果 krb5.conf 文件格式有问题，关联的服务很容易受到攻击。您必须首先解决该问题，然后再允许使用 Kerberos 功能。

传播 Kerberos 数据库时出现问题

如果 Kerberos 数据库传播失败，请在从 KDC 与主 KDC 服务器之间尝试使用 /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该问题。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 无法正常运行，可能是因为 KDC 上的密钥表文件存在问题。如果 rlogin 可以正常运行，则问题不在于密钥表文件或名称服务，因为 rlogin 与传播软件使用同一个 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。如果文件系统不是由 root 所有，请删除该文件系统并再次尝试挂载。
- 如果访问基于 Kerberos 的 NFS 文件系统时出现问题，请确保您的系统和 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，可能是因为在尝试挂载 NFS 文件系统时使用的不是全限定 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

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 用户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
```

接下来，指示 `gssd` 服务从 `/etc/gss/gsscred.conf` 文件获取信息。

```
# 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。

¹ Ken Hornstein, "Kerberos FAQ", [<http://www.cmf.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html#kerbdns>], 访问时间 2010 年 3 月 10 日。

对 Kerberos 服务使用 DTrace

在本示例中，您想知道 KDC 是否要求预验证，如果要求，支持哪些预验证类型。首先，以特权用户的身份创建一个 D 程序源文件，如下所示：

```
# cat kerberos_preauth.d
kerberos$target::krb_error-read
{
    self->preauth = args[1]->kerror_error_code ==
        "KDC_ERR_PREAUTH_REQUIRED(25)" ? "required" : "not required";

    printf(" - Preauthentication is %s for this KDC.\n", self->preauth);
}

kerberos$target::krb_error-read
/ self->preauth == "required" /
{
    printf(" - This KDC supports the following preauth types: %s.",
        args[1]->kerror_e_data);
}
```

接下来，编译 preauth.d 源文件以获得答案。

```
# dtrace -qs kerberos_preauth.d -c "kinit -k"
- Preauthentication is required for this KDC.
- This KDC supports the following preauth types: ENC_TIMESTAMP(2)
FX_FAST(136) PK_ETYPE_INFO2(19) SAM_RESPONSE(13) FX_COOKIE(133).
```

有关各种预验证类型的更多信息，请参见 [RFC 4120](#)。

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所有用户都应该阅读本章。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 19 章，[Kerberos 服务介绍](#)和第 20 章，[规划 Kerberos 服务](#)。

以下是本章中信息的列表。

- 第 411 页中的“管理 Kerberos 主体和策略的方法”
- 第 412 页中的“SEAM Tool”
- 第 416 页中的“管理 Kerberos 主体”
- 第 429 页中的“管理 Kerberos 策略”
- 第 437 页中的“SEAM Tool 参考”
- 第 441 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命令行接口，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，Oracle Solaris 发行版还提供了 SEAM Tool (gkadmin)，这是一个交互式图形用户界面 (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 412 页中的“SEAM Tool”。

SEAM Tool

SEAM Tool (gkadmin) 是一个交互式图形用户界面 (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 441 页中的“管理密钥表文件”中所述。

与 kadmin 命令类似，SEAM Tool 使用 Kerberos 验证和加密的 RPC 从网络中的任何位置进行安全操作。使用 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 还会提供上下文有关帮助和一般联机帮助。

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 416 页中的“管理 Kerberos 主体（任务列表）”
- 第 429 页中的“管理 Kerberos 策略（任务列表）”

此外，还可转至第 437 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，这些命令提供的功能与 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 23-1 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

表 23-1 SEAM Tool 的等效命令行 (续)

SEAM Tool 过程	等效的 kadmin 命令
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
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" (编辑) 菜单中选择 "Properties" (属性)，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" (打印) 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
- 指定主体 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" (帮助) 菜单，可以访问上下文相关帮助和一般帮助。从 "Help" (帮助) 菜单中选择 "Context-Sensitive Help" (上下文相关帮助) 时，将显示 "Context-Sensitive Help" (上下文相关帮助) 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" (帮助) 窗口中显示有关该项目的帮助。要切换回工具的正常模式，请在 "Help" (帮助) 窗口中单击 "Dismiss" (解除)。

此外，还可选择 "Help Contents" (帮助内容)，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"（编辑）菜单中选择 "Properties"（属性），然后取消选中 "Show Lists"（显示列表）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。相应的做法是，必须在提供的新 "Name"（名称）字段中键入主体或策略名称，然后选择要对其执行的操作。实际上，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。事实上，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"（编辑）菜单中选择 "Properties"（属性），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；也可以指定该工具必须将列表从服务器重新装入高速缓存的时间限制。

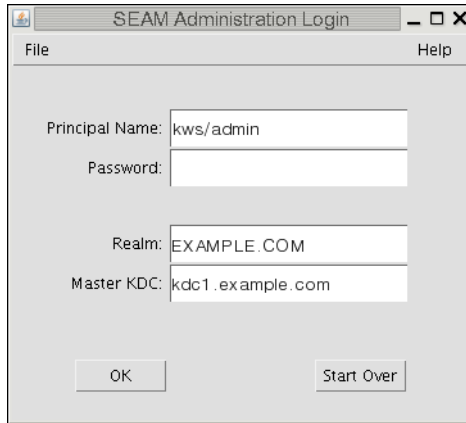
对列表进行高速缓存时，仍然可以使用列表面板来选择主体和策略，因此该方法不会像第一种方法那样影响 SEAM Tool 的使用方式。另外，尽管使用高速缓存使您无法查看其他用户所做的更改，但您仍可以根据自己所做的更改查看最新列表信息，因为您所做的更改会对服务器和高速缓存中的列表进行更新。而且，如果要更新高速缓存以查看其他更改并获取最新列表副本，可在需要从服务器刷新高速缓存时使用 "Refresh"（刷新）菜单。

▼ 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" (SEAM 管理登录) 窗口。



- 2 如果不想使用这些缺省值，请指定新的缺省值。

该窗口自动使用缺省值进行填充。缺省主体名称 (`username/admin`) 是通过从 `USER` 环境变量获取当前标识并在其后附加 `/admin` 确定的。缺省的 "Realm" (领域) 和 "Master KDC" (主 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" (重新开始)。

注 - 每个主体名称可以执行的管理操作是由 Kerberos ACL 文件 `/etc/krb5/kadm5.acl` 指定的。有关受限特权的信息，请参见第 440 页中的“[以受限 Kerberos 管理特权使用 SEAM Tool](#)”。

- 3 键入指定主体名称的口令。
- 4 单击“确定”。

此时会显示包含所有主体的窗口。

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"（主体）选项卡来查看主体列表。	第 417 页中的“如何查看 Kerberos 主体列表”
查看主体属性。	通过在 "Principal List"（主体列表）中选择 "Principal"（主体），然后单击 "Modify"（修改）按钮来查看主体的属性。	第 419 页中的“如何查看 Kerberos 主体属性”
创建新主体。	通过单击 "Principal List"（主体列表）面板中的 "Create New"（新建）按钮来创建新主体。	第 421 页中的“如何创建新的 Kerberos 主体”
复制主体。	通过在 "Principal List"（主体列表）中选择要复制的主体，然后单击 "Duplicate"（复制）按钮来复制主体。	第 424 页中的“如何复制 Kerberos 主体”
修改主体。	通过在 "Principal List"（主体列表）中选择要修改的主体，然后单击 "Modify"（修改）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 424 页中的“如何修改 Kerberos 主体”
删除主体。	通过在 "Principal List"（主体列表）中选择要删除的主体，然后单击 "Delete"（删除）按钮来删除主体。	第 425 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"（编辑）菜单中选择 "Properties"（属性）来设置缺省值以创建新主体。	第 426 页中的“如何设置缺省值以创建新的 Kerberos 主体”
修改 Kerberos 管理特权（kadm5.acl 文件）。	仅限命令行。 Kerberos 管理特权确定主体可对 Kerberos 数据库执行的操作，如添加和修改。 要修改每个主体的 Kerberos 管理特权，需要编辑 /etc/krb5/kadm5.acl 文件。	第 427 页中的“如何修改 Kerberos 管理特权”

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。该脚本将读入一个名为 `princnames` 的文件（其中包含主体名称及其口令），然后将这些主体名称和口令添加至 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。

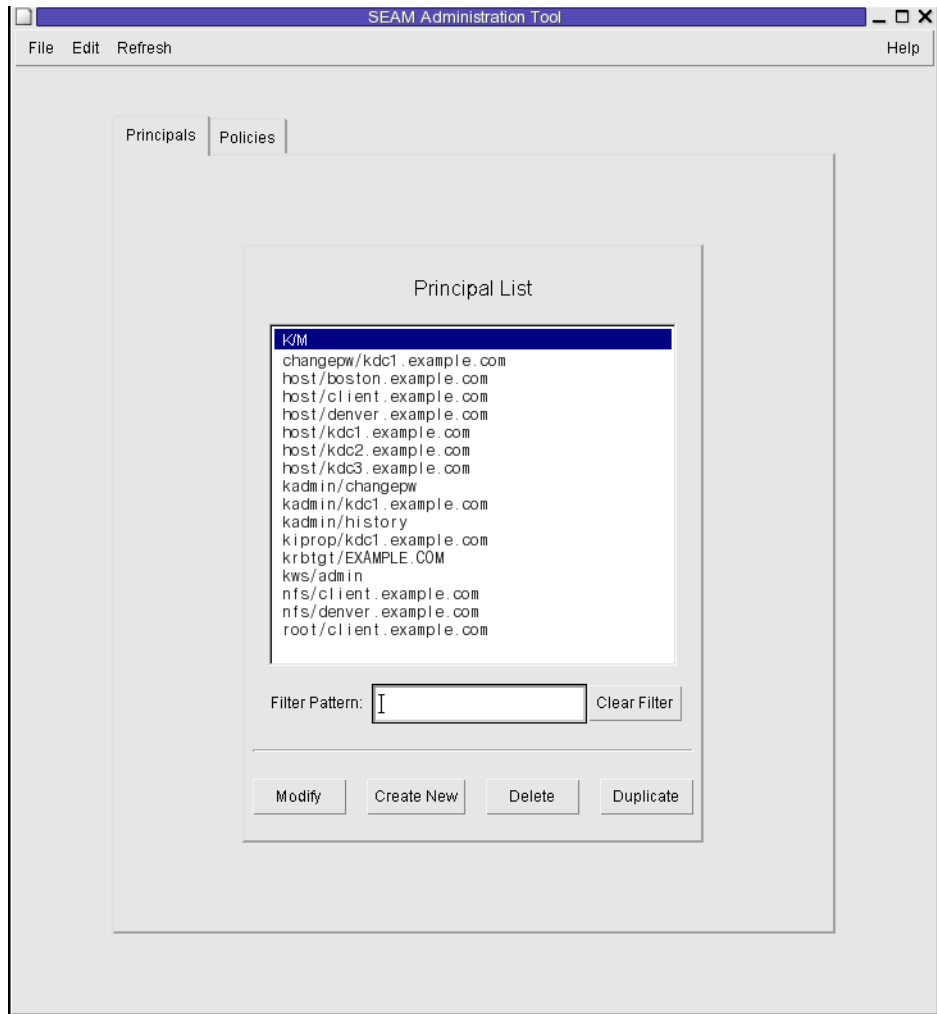
1 如有必要，启动 SEAMTool。

有关更多信息，请参见第 415 页中的“如何启动 SEAM Tool”。

```
$ /usr/sbin/gkadmin
```

2 单击 "Principals" (主体) 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" (过滤器) 字段中键入过滤字符串，然后按回车键。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，如果键入了过滤字符串 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"（清除过滤器）。

示例 23-1 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 `kadmin*` 匹配的所有主体。通配符可与 `list_principals` 命令一起使用。

```
kadmin: list_principals kadmin*
kadmin/changepw@EXAMPLE.COM
kadmin/kdc1.example.con@EXAMPLE.COM
kadmin/history@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 `SEAMTool`。

有关更多信息，请参见第 415 页中的“如何启动 `SEAM Tool`”。

```
$ /usr/sbin/gkadmin
```

- 2 单击 "Principals"（主体）选项卡。

- 3 在列表中选择要查看的主体，然后单击 "Modify"（修改）。

此时会显示包含该主体某些属性的 "Principal Basics"（主体基本信息）面板。

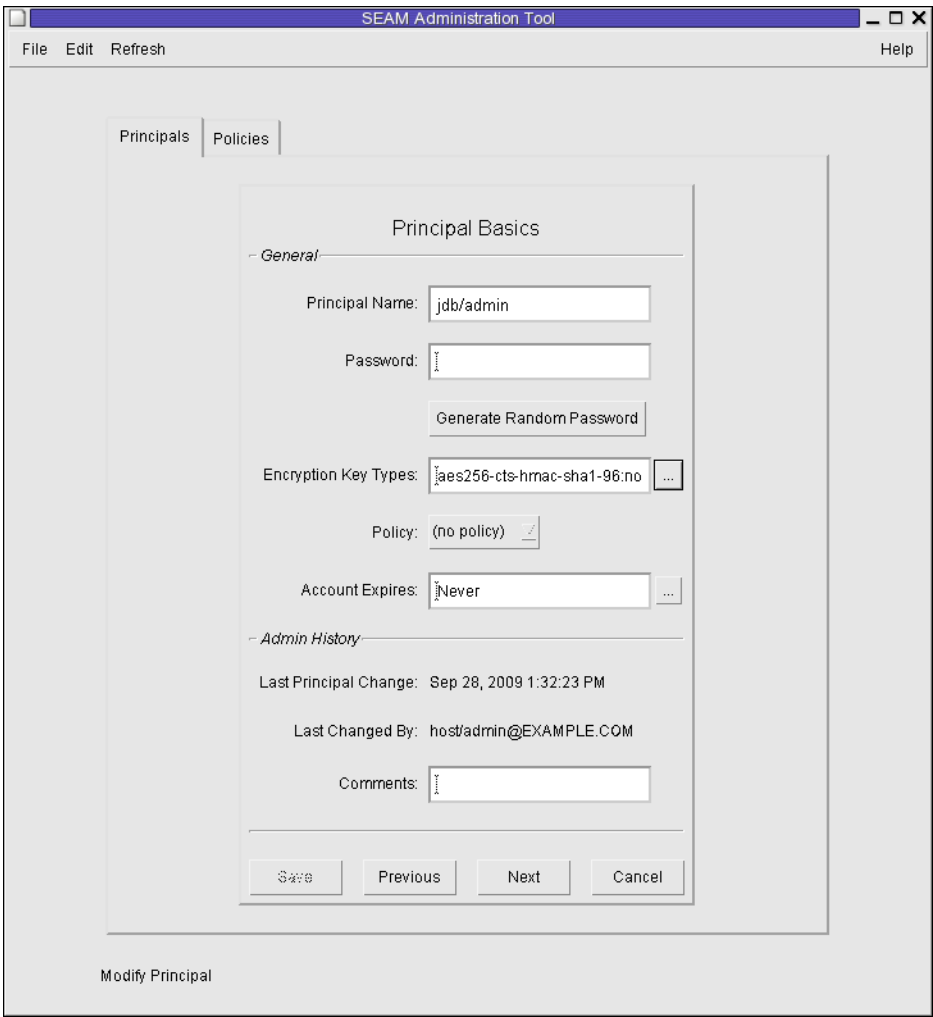
- 4 继续单击 "Next"（下一页）以查看该主体的所有属性。

有三个窗口包含属性信息。请从 "Help"（帮助）菜单中选择 "Context-Sensitive Help"（上下文相关帮助），获取有关每个窗口中各种属性的信息。或者，转至第 437 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"（取消）。

示例 23-2 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 23-3 查看 Kerberos 主体属性（命令行）

在以下示例中，使用 `kadmin` 的 `get_principal` 命令来查看 `jdb/admin` 主体的属性。

```
kadmin: getprinc jdb/admin
Principal: jdb/admin@EXAMPLE.COM

Expiration date: [never]
Last password change: [never]

Password expiration date: Wed Apr 14 11:53:10 PDT 2011
Maximum ticket life: 1 day 16:00:00
Maximum renewable life: 1 day 16:00:00
Last modified: Mon Sep 28 13:32:23 PST 2009 (host/admin@EXAMPLE.COM)
```



```

Last successful authentication: [never]
Last failed authentication: [never]
Failed password attempts: 0
Number of keys: 1
Key: vno 1, AES-256 CTS mode with 96-bit SHA-1 HMAC, no salt
Key: vno 1, AES-128 CTS mode with 96-bit SHA-1 HMAC, no salt
Key: vno 1, Triple DES with HMAC/sha1, no salt
Key: vno 1, ArcFour with HMAC/md5, no salt
Key: vno 1, DES cbc mode with RSA-MD5, 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 SEAMTool。

有关更多信息，请参见第 415 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 433 页中的“如何创建新的 Kerberos 策略”。

```
$ /usr/sbin/gkadmin
```

2 单击 "Principals" (主体) 选项卡。

3 单击 "New" (新建)。

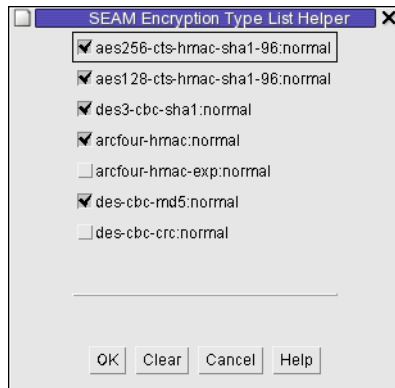
此时会显示包含某些主体属性的 "Principal Basics" (主体基本信息) 面板。

4 指定主体名称和口令。

必须提供主体名称和口令。

5 指定该主体的加密类型。

单击加密密钥类型字段右侧的框，以打开一个新窗口，其中将显示所有可用加密密钥类型。选择所需的加密类型后，单击 "OK"（确定）。



6 指定该主体的策略。

7 指定主体属性的值，并继续单击 "Next"（下一页）以指定更多属性。

有三个窗口包含属性信息。请从 "Help"（帮助）菜单中选择 "Context-Sensitive Help"（上下文相关帮助），获取有关每个窗口中各种属性的信息。或者，转至第 437 页中的“SEAM Tool 面板说明”，了解所有主体属性说明。

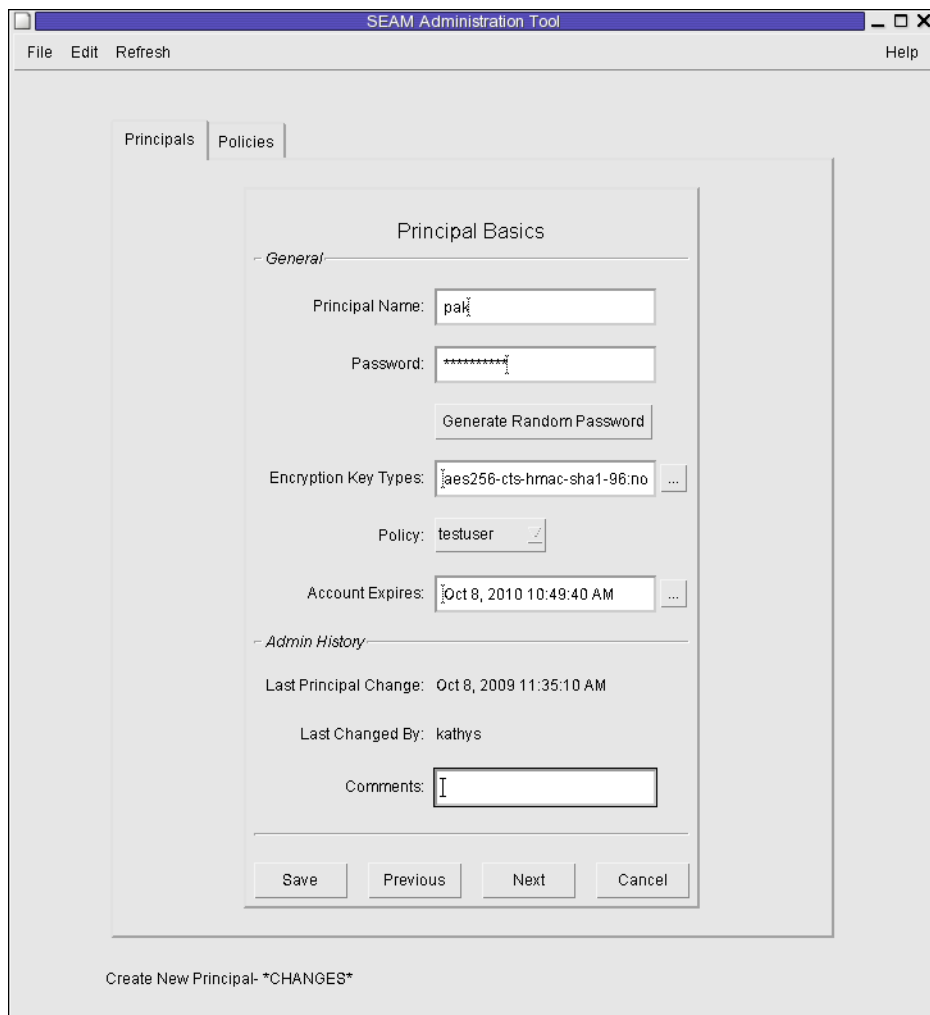
8 单击 "Save"（保存）以保存主体，或在最后一个面板上单击 "Done"（完成）。

9 如有必要，在 /etc/krb5/kadm5.acl 文件中为新主体设置 Kerberos 管理特权。

有关更多详细信息，请参见第 427 页中的“如何修改 Kerberos 管理特权”。

示例 23-4 创建新的 Kerberos 主体

以下示例显示了创建名为 pak 的新主体时的 "Principal Basics"（主体基本信息）面板。策略设置为 testuser。



示例 23-5 创建新的 Kerberos 主体（命令行）

在以下示例中，使用 `kadmin` 的 `add_principal` 命令来创建一个名为 `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 SEAMTool。

有关更多信息，请参见第 415 页中的“如何启动 SEAM Tool”。

```
$ /usr/sbin/gkadmin
```

- 2 单击 "Principals" (主体) 选项卡。

- 3 在列表中选择要复制的主体，然后单击 "Duplicate" (复制)。

此时会显示 "Principal Basics" (主体基本信息) 面板。除空的 "Principal Name" (主体名称) 和 "Password" (口令) 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" (保存) 并跳至步骤 7。

- 5 指定主体属性的其他值，然后继续单击 "Next" (下一页) 以指定更多属性。

有三个窗口包含属性信息。请从 "Help" (帮助) 菜单中选择 "Context-Sensitive Help" (上下文相关帮助)，获取有关每个窗口中各种属性的信息。或者，转至第 437 页中的“SEAM Tool 面板说明”，了解所有主体属性说明。

- 6 单击 "Save" (保存) 以保存主体，或在最后一个面板上单击 "Done" (完成)。

- 7 如有必要，在 /etc/krb5/kadm5.acl 文件中为主体设置 Kerberos 管理特权。

有关更多详细信息，请参见第 427 页中的“如何修改 Kerberos 管理特权”。

▼ 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 SEAMTool。

有关更多信息，请参见第 415 页中的“如何启动 SEAM Tool”。

```
$ /usr/sbin/gkadmin
```

- 2 单击 "Principals" (主体) 选项卡。

- 3 在列表中选择要修改的主体，然后单击 "Modify" (修改)。

此时会显示包含该主体某些属性的 "Principal Basics" (主体基本信息) 面板。

- 4 修改该主体的属性，并继续单击 "Next"（下一页）以修改其他属性。

有三个窗口包含属性信息。请从 "Help"（帮助）菜单中选择 "Context-Sensitive Help"（上下文相关帮助），获取有关每个窗口中各种属性的信息。或者，转至第 437 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"（保存）以保存主体，或在最后一个面板上单击 "Done"（完成）。
- 6 在 `/etc/krb5/kadm5.acl` 文件中，修改该主体的 Kerberos 管理特权。
有关更多详细信息，请参见第 427 页中的“如何修改 Kerberos 管理特权”。

示例 23-6 修改 Kerberos 主体的口令（命令行）

在以下示例中，使用 `kadmin` 的 `change_password` 命令来修改 `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 415 页中的“如何启动 SEAM Tool”。
`$ /usr/sbin/gkadmin`
- 2 单击 "Principals"（主体）选项卡。
- 3 在列表中选择要删除的主体，然后单击 "Delete"（删除）。
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 427 页中的“如何修改 Kerberos 管理特权”。

示例 23-7 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

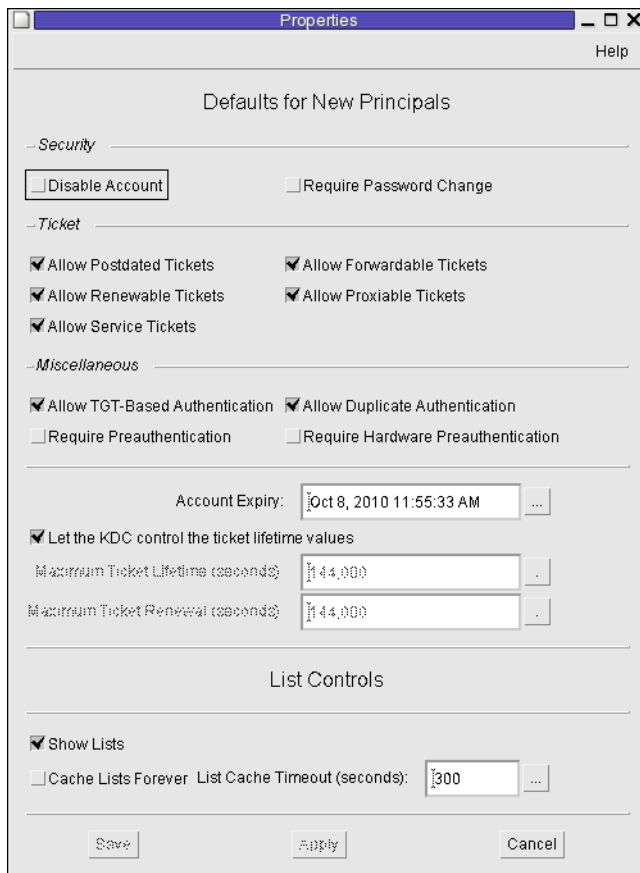
此过程没有等效命令行。

1 如有必要，启动 SEAMTool。

有关更多信息，请参见第 415 页中的“如何启动 SEAM Tool”。

```
$ /usr/sbin/gkadmin
```

- 2 从 "Edit" (编辑) 菜单中选择 "Properties" (属性)。
此时会显示 "Properties" (属性) 窗口。



- 3 选择要在创建新主体时使用的缺省值。

请从 "Help" (帮助) 菜单中选择 "Context-Sensitive Help" (上下文相关帮助)，获取有关每个窗口中各种属性的信息。

- 4 单击“保存”。

▼ 如何修改 Kerberos 管理特权

尽管您的站点可能有許多用户主体，但您通常只希望一小部分用户能够管理 Kerberos 数据库。管理 Kerberos 数据库的特权由 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特权。或者，可在主体名称中使用 "*" 通配符来指定主体组的特权。

1 成为主 KDC 服务器上的超级用户。

2 编辑 `/etc/krb5/kadm5.acl` 文件。

`kadm5.acl` 文件中的条目必须采用以下格式：

principal privileges [principal-target]

principal 指定要为其授予特权的主体。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特权时很有用。例如，如果要指定包含 `admin` 实例的所有主体，需要使用 `*/admin@realm`。

请注意，`admin` 实例常用于将单独的特权（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特权时，才会获取 `jdb/admin` 票证。

privileges 指定主体能够执行或不能执行的操作。此字段由一个或多个以下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 * 允许所有特权 (admcil)。

principal-target 当在此字段中指定了某个主体时，*privileges* 仅在 *principal* 对 *principal-target* 进行操作时，才会应用于 *principal*。主体名称的任何部分都可以包含 "*" 通配符，这在主体分组时很有用。

示例 23-8 修改 Kerberos 管理特权

`kadm5.acl` 文件中的以下条目授予 `EXAMPLE.COM` 领域中包含 `admin` 实例的任何主体对 Kerberos 数据库的所有特权：

`*/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` 实例的任何主体的特权。

`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"（策略）选项卡来查看策略列表。	第 429 页中的“如何查看 Kerberos 策略列表”
查看策略属性。	通过在 "Policy List"（策略列表）中选择策略，然后单击 "Modify"（修改）按钮来查看策略的属性。	第 431 页中的“如何查看 Kerberos 策略属性”
创建新策略。	通过单击 "Policy List"（策略列表）面板中的 "Create New"（新建）按钮来创建新策略。	第 433 页中的“如何创建新的 Kerberos 策略”
复制策略。	通过在 "Policy List"（策略列表）中选择要复制的策略，然后单击 "Duplicate"（复制）按钮来复制策略。	第 435 页中的“如何复制 Kerberos 策略”
修改策略。	通过在 "Policy List"（策略列表）中选择要修改的策略，然后单击 "Modify"（修改）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 435 页中的“如何修改 Kerberos 策略”
删除策略。	通过在 "Policy List"（策略列表）中选择要删除的策略，然后单击 "Delete"（删除）按钮来删除策略。	第 436 页中的“如何删除 Kerberos 策略”

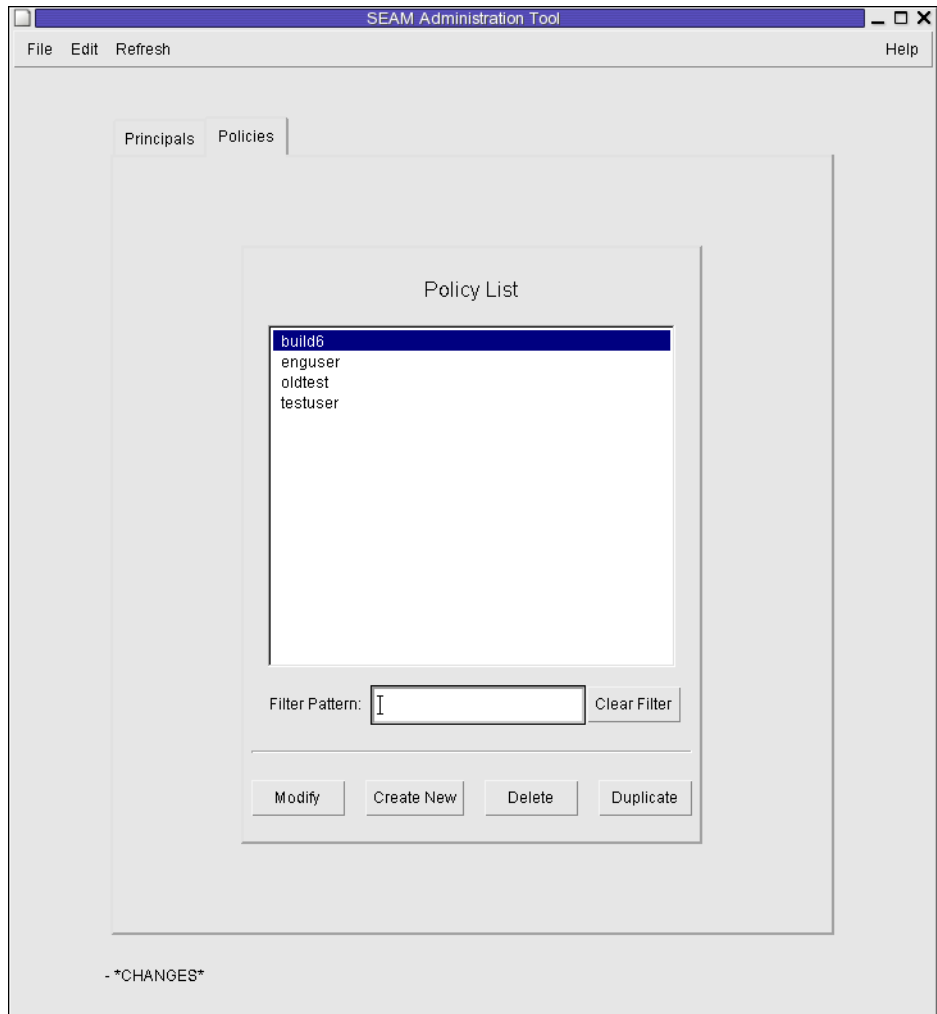
▼ 如何查看 Kerberos 策略列表

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 415 页中的“如何启动 SEAM Tool”](#)。
- \$ /usr/sbin/gkadmin

2 单击 "Policies" (策略) 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" (过滤器) 字段中键入过滤字符串，然后按回车键。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，如果键入了过滤字符串 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"（清除过滤器）。

示例 23-9 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 415 页中的“如何启动 SEAM Tool”。
- 2 单击 "Policies"（策略）选项卡。

```
$ /usr/sbin/gkadmin
```
- 3 在列表中选择要查看的策略，然后单击 "Modify"（修改）。
此时会显示 "Policy Details"（策略详细信息）面板。
- 4 查看完毕后，单击 "Cancel"（取消）。

示例 23-10 查看 Kerberos 策略属性

以下示例显示了查看 `test` 策略时的 "Policy Details"（策略详细信息）面板。



示例 23-11 查看 Kerberos 策略属性（命令行）

在以下示例中，使用 `kadmin` 的 `get_policy` 命令来查看 `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
```

Reference count（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 SEAMTool。

有关更多信息，请参见第 415 页中的“如何启动 SEAM Tool”。

```
$ /usr/sbin/gkadmin
```

- 2 单击 "Policies"（策略）选项卡。

- 3 单击 "New"（新建）。

此时会显示 "Policy Details"（策略详细信息）面板。

- 4 在 "Policy Name"（策略名称）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"（帮助）菜单中选择 "Context-Sensitive Help"（上下文相关帮助），获取有关此窗口中各种属性的信息。或者，转至表 23-5，了解所有策略属性说明。

- 6 单击 "Save"（保存）按钮以保存策略，或单击 "Done"（完成）。

示例 23-12 创建新的 Kerberos 策略

在以下示例中，创建了一个名为 build11 的新策略。"Minimum Password Classes"（最少口令类）设置为 3。



示例 23-13 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 SEAMTool。

有关更多信息，请参见第 415 页中的“如何启动 SEAM Tool”。

```
$ /usr/sbin/gkadmin
```

- 2 单击 "Policies" (策略) 选项卡。

- 3 在列表中选择要复制的策略，然后单击 "Duplicate" (复制)。

此时会显示 "Policy Details" (策略详细信息) 面板。除空的 "Policy Name" (策略名称) 字段之外，选定策略的其他属性都将被复制。

- 4 在 "Policy Name" (策略名称) 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" (帮助) 菜单中选择 "Context-Sensitive Help" (上下文相关帮助)，获取有关此窗口中各种属性的信息。或者，转至表 23-5，了解所有策略属性说明。

- 6 单击 "Save" (保存) 按钮以保存策略，或单击 "Done" (完成)。

▼ 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 SEAMTool。

有关详细信息，请参见第 415 页中的“如何启动 SEAM Tool”。

```
$ /usr/sbin/gkadmin
```

- 2 单击 "Policies" (策略) 选项卡。

- 3 在列表中选择要修改的策略，然后单击 "Modify" (修改)。

此时会显示 "Policy Details" (策略详细信息) 面板。

- 4 修改该策略的属性。

请从 "Help" (帮助) 菜单中选择 "Context-Sensitive Help" (上下文相关帮助)，获取有关此窗口中各种属性的信息。或者，转至表 23-5，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" (保存) 按钮以保存策略，或单击 "Done" (完成) 。

示例 23-14 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五个字符。

```
$ 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 Policy (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 SEAMTool。
有关更多信息，请参见第 415 页中的“如何启动 SEAM Tool”。
 - 2 单击 "Policies" (策略) 选项卡。
 - 3 在列表中选择要删除的策略，然后单击 "Delete" (删除) 。
- 确认删除后，将删除该策略。

示例 23-15 删除 Kerberos 策略 (命令行)

在以下示例中，使用 kadmin 的 delete_policy 命令来删除 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。如果某个主体正在使用该策略，则 delete_policy 命令会失败。

SEAMTool 参考

本节介绍了 SEAM Tool 中的各个面板。此外，还提供了有关以受限特权使用 SEAM Tool 的信息。

SEAMTool 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 23-2 SEAM Tool 的 "Principal Basics"（主体基本信息）面板中的属性

属性	说明
Principal Name（主体名称）	主体的名称（全限定主体名称的 <i>primary/instance</i> 部分）。主体是 KDC 可以为其分配票证的唯一标识。 修改主体时不能编辑其名称。
Password（口令）	主体的口令。可使用 "Generate Random Password"（生成随机口令）按钮为主体创建随机口令。
Policy（策略）	主体的可用策略菜单。
Account Expires（帐户过期）	主体帐户的过期日期和时间。帐户过期后，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（上次主体更改）	上次修改主体信息的日期。（只读）
Last Changed By（上次更改者）	上次修改此主体帐户的主体的名称。（只读）
Comments（注释）	与主体有关的注释（如“临时帐户”）。

表 23-3 SEAM Tool 的 "Principal Details"（主体详细信息）面板中的属性

属性	说明
Last Success（上次成功）	主体上次登录成功的日期和时间。（只读）
Last Failure（上次失败）	主体上次登录失败的日期和时间。（只读）
Failure Count（失败计数）	主体登录失败的次数。（只读）
Last Password Change（上次口令更改）	上次更改主体口令的日期和时间。（只读）
Password Expires（口令过期）	主体当前口令过期的日期和时间。
Key Version（密钥版本）	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。

表 23-3 SEAM Tool 的 "Principal Details" (主体详细信息) 面板中的属性 (续)

属性	说明
Maximum Lifetime (seconds) (最长生命周期 (秒))	可将票证授予主体的最长时间 (不更新)。
Maximum Renewal (seconds) (最长更新期 (秒))	主体可更新现有票证的最长时间。

表 23-4 SEAM Tool 的 "Principal Flags" (主体标志) 面板中的属性

属性 (单选按钮)	说明
Disable Account (禁用帐户)	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 (要求口令更改)	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 <code>kpasswd</code> 命令来创建新口令。如果发生安全违规, 并且需要确保替换旧口令, 则此属性很有用。
Allow Postdated Tickets (允许以后生效的票证)	选中此项后, 将允许主体获取以后生效的票证。 例如, 如果 <code>cron</code> 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 (允许可转发票证)	选中此项后, 将允许主体获取可转发的票证。 可转发票证是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 <code>ftp</code> 或 <code>rsh</code> 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 (允许可更新票证)	选中此项后, 将允许主体获取可更新的票证。 主体可以自动延长可更新票证的失效日期或时间, 而不必在票证首次过期后获取新的票证。目前, NFS 服务是可以更新票证的票证服务。
Allow Proxiable Tickets (允许可代理票证)	选中此项后, 将允许主体获取可代理的票证。 可代理票证是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 (允许服务票证)	选中此项后, 将允许为主体颁发服务票证。 不应允许为 <code>kadmin/hostname</code> 和 <code>changepw/hostname</code> 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 (允许基于 TGT 的验证)	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。

表 23-4 SEAM Tool 的 "Principal Flags" (主体标志) 面板中的属性 (续)

属性 (单选按钮)	说明
Allow Duplicate Authentication (允许复制验证)	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。
Required Preauthentication (要求预先验证)	选中此项后, KDC 在验证 (通过软件) 主体确实为请求 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 (例如, 从 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 (要求硬件验证)	选中此项后, KDC 在验证 (通过硬件) 主体确实为请求 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 23-5 SEAM Tool 的 "Policy Basics" (策略基本信息) 面板中的属性

属性	说明
Policy Name (策略名称)	策略的名称。策略是用于管理主体口令和票证的规则集合。 修改策略时不能编辑其名称。
Minimum Password Length (最小口令长度)	主体口令的最小长度。
Minimum Password Classes (最少口令类)	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 (保存的口令历史记录)	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) (最短口令生命周期 (秒))	口令在可更改之前必须经历的最短时间。
Maximum Password Lifetime (seconds) (最长口令生命周期 (秒))	口令在必须更改之前可以经历的最长时间。
Principals Using This Policy (使用此策略的主体)	当前应用此策略的主体数。(只读)

以受限 Kerberos 管理特权使用 SEAM Tool

如果 admin 主体拥有管理 Kerberos 数据库的所有特权，则可使用 SEAM Tool 的所有功能。不过，您的特权可能受到限制，如仅允许查看主体列表或更改主体口令。借助受限 Kerberos 管理特权，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 管理特权而变化。表 23-6 显示了 SEAM Tool 基于 Kerberos 管理特权变化的情况。

没有列表特权时，SEAM Tool 会发生最直观的变化。如果没有列表特权，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"（名称）字段来指定要处理的主体或策略。

如果您登录到 SEAM Tool，但却没有足够的特权来使用它执行任务，则会显示以下消息并且会返回到 "SEAM Administration Login"（SEAM 管理登录）窗口：

Insufficient privileges to use gkadmin: ADMCIL. Please try using another principal.

要更改主体的特权以便它可管理 Kerberos 数据库，请转至第 427 页中的“如何修改 Kerberos 管理特权”。

表 23-6 以受限 Kerberos 管理特权使用 SEAM Tool

禁用的特权	SEAM Tool 如何变化
a（添加）	"Principal List"（主体列表）和 "Policy List"（策略列表）面板中的 "Create New"（新建）和 "Duplicate"（复制）按钮不可用。如果没有添加特权，则无法创建新主体或策略，也不能复制它们。
d（删除）	"Principal List"（主体列表）和 "Policy List"（策略列表）面板中的 "Delete"（删除）按钮不可用。如果没有删除特权，则无法删除主体或策略。
m（修改）	"Principal List"（主体列表）和 "Policy List"（策略列表）面板中的 "Modify"（修改）按钮不可用。如果没有修改特权，则无法修改主体或策略。 而且，如果 "Modify"（修改）按钮不可用，则即使您拥有更改口令的特权，也不能修改主体的口令。
c（更改口令）	"Principal Basics"（主体基本信息）面板中的 "Password"（口令）字段处于只读状态，无法更改。如果没有更改口令的特权，则无法修改主体的口令。 请注意，即使您拥有更改口令的特权，还必须同时拥有修改特权才能更改主体的口令。

表 23-6 以受限 Kerberos 管理特权使用 SEAM Tool (续)

禁用的特权	SEAM Tool 如何变化
i (查询数据库)	"Principal List" (主体列表) 和 "Policy List" (策略列表) 面板中的 "Modify" (修改) 和 "Duplicate" (复制) 按钮不可用。如果没有查询特权, 则无法修改或复制主体或策略。 而且, 如果 "Modify" (修改) 按钮不可用, 则即使您拥有更改口令的特权, 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。如果没有列表特权, 则必须使用列表面板中的 "Name" (名称) 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一个名为 *keytab* 的本地文件, *keytab* 是“key table”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。在提供基于 Kerberos 的服务的应用服务器上, 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。此外, 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* 命令来获取客户机的 *root* 主体的凭证。

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 管理特权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 <code>kadmin</code> 的 <code>ktadd</code> 命令将服务主体添加至密钥表文件。	第 442 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 <code>kadmin</code> 的 <code>ktremove</code> 命令从密钥表文件中删除服务。	第 443 页中的“如何从密钥表文件中删除服务主体”
显示密钥表文件中的密钥列表（主体列表）。	使用 <code>ktutil</code> 命令来显示密钥表文件中的密钥列表。	第 444 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 <code>kadmin</code> 特权。 使用 <code>ktutil</code> 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 445 页中的“如何在主机上临时禁用对服务的验证”

▼ 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 417 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 23-16 将服务主体添加至密钥表文件

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 可以验证 `denver` 的网络服务。

```
denver # /usr/sbin/kadmin
kadmin: ktadd host/denver.example.com
Entry for principal host/denver.example.com with kvno 3, encryption type AES-256 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 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 444 页中的“如何显示密钥表文件中的密钥列表（主体）”。
- 4 使用 `ktremove` 命令从密钥表文件删除主体。

```
kadmin: ktrremove [-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 23-17 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** 所有权。

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 23-18 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
```

▼ 如何在主机上临时禁用对服务的验证

有时可能需要在网络应用服务器上，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` 特权。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` 所有权。

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。请记住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 如果要重新启用该服务，请将临时（原始）密钥表文件复制回其原始位置。**示例 23-19 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 文件复制为 /etc/krb5/krb5.keytab 文件。

```
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
```

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。本章说明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 447 页中的“Kerberos 票证管理”](#)
- [第 450 页中的“Kerberos 口令管理”](#)
- [第 455 页中的“Kerberos 用户命令”](#)

有关 Oracle Solaris Kerberos 产品的概述，请参见[第 19 章，Kerberos 服务介绍](#)。

Kerberos 票证管理

本节说明如何获取、查看和销毁票证。有关票证的介绍，请参见[第 302 页中的“Kerberos 服务的工作原理”](#)。

是否需要担心票证？

安装任何 SEAM 发行版或 Oracle Solaris 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、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 455 页中的“基于 Kerberos 的命令概述”](#)和[第 457 页中的“转发 Kerberos 票证”](#)。

有关票证生命周期的信息，请参见[第 466 页中的“票证生命周期”](#)。

创建 Kerberos 票证

通常，如果 PAM 配置正确，登录时会自动创建票证，您无需执行任何特殊操作即可获得票证。但是，如果票证过期，则可能需要创建票证。另外，除了缺省主体外，您可能还需要使用另一个主体。例如，在使用 `rlogin -l` 以其他用户身份登录到计算机时。

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 24-1 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>
```

此示例展示了用户 `david` 如何使用 `-f` 选项为自己创建可转发票证。例如，该用户可以使用此可转发票证登录到第二个系统，然后通过 `telnet` 远程登录到第三个系统。

```
% kinit -f david@EXAMPLE.ORG
Password for david@EXAMPLE.ORG:      <Type password>
```

有关可转发票证工作原理的更多信息，请参见第 457 页中的“转发 Kerberos 票证”和第 465 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 *forwardable* 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 Preauthenticated（已预验证）

D Postdateable (可以后生效)
 d Postdated (以后生效)
 F Forwardable (可转发)
 f Forwarded (已转发)
 I Initial (初始)
 i Invalid (无效)
 P Proxiabile (可代理)
 p Proxy (代理)
 R Renewable (可更新)

第 465 页中的“票证类型”介绍了票证可以具有的各种属性。

示例 24-2 查看 Kerberos 票证

本示例说明用户 *jennifer* 拥有一个 *initial* 票证，该票证是 *forwardable* (F) 和 *postdated* (d) 票证，但尚未经过验证 (i)。

```
% /usr/bin/klist -f
Ticket cache: /tmp/krb5cc_74287
Default principal: jennifer@EXAMPLE.COM

Valid starting          Expires              Service principal
09 Mar 04 15:09:51      09 Mar 04 21:09:51  nfs/EXAMPLE.COM@EXAMPLE.COM
        renew until 10 Mar 04 15:12:51, Flags: Fdi
```

以下示例说明 *david* 有两个票证，这些票证是从另一台主机转发 (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.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为主机名。

```
% klist -fea
Ticket cache: /tmp/krb5cc_74287
```

示例 24-2 查看 Kerberos 票证 (续)

```
Default principal: david@EXAMPLE.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可能性。

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`，或使用屏幕保护程序来锁定屏幕。

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。可以将两个口令设置为相同或不同。

关于选择口令的建议

口令几乎可以包括您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
- 著名人物或角色的姓名
- 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
- 您的身份证号、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 "RrriotGrrrlsRrrrule!"



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，`passwd` 命令还会自动提示您输入新的 Kerberos 口令。

使用 `passwd` 而非 `kpasswd` 的好处在于，您可以同时设置 UNIX 口令和 Kerberos 口令。但是，在一般情况下，**不必**使用 `passwd` 同时更改这两个口令。通常，您可以只更改 UNIX 口令而保持 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，而对于另一些站点，则要求更改 Kerberos 口令。

- 使用 `kpasswd` 命令。`kpasswd` 与 `passwd` 非常相似。这两个命令的一个区别是，`kpasswd` 只能更改 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将需要一些时间（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即获取新的 Kerberos 票证，请先尝试新口令。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 429 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 24-3 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 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。

此示例展示了用户 `david` 如何使用 `kpasswd` 仅更改其 Kerberos 口令。

示例 24-3 更改口令 (续)

```
% 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：

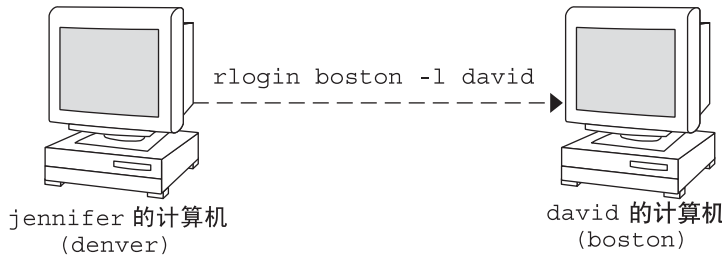
```
jennifer@ENG.EXAMPLE.COM
joe@EXAMPLE.ORG
```

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 的身份远程登录到的计算机 (boston)，而不必提供 david 的口令。

图 24-1 使用 .k5login 文件授予对帐户的访问权限

jennifer 可以用 david 的帐户登录到 david 的计算机，而无需输入 david 的口令。

david 有一个 .k5login 文件，此文件中包含 jennifer@ENG.ACME.COM



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 448 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 24-4 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
```

示例 24-4 使用 .k5login 文件授予对帐户的访问权限 (续)

```
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
- 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和可选加密的功能。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）。
- 加密您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。有关此处所述命令的详细说明，请参见各自的手册页。

下列 Kerberos 选项已添加到 ftp、rcp、rlogin、rsh 和 telnet：

- a 尝试使用现有票证自动登录。使用 `getlogin()` 返回的用户名，除非该名称与当前用户 ID 不相同。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。不能在同一个命令中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 457 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。不能在同一个命令中同时使用这两个选项。

有关转发票证的更多信息，请参见第 457 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* 指定要使用的 GSS-API 安全机制，如 `/etc/gss/mech` 文件中所列。缺省值为 `kerberos_v5`。
- x 加密此会话。

-X *auth-type* 禁用 *auth-type* 类型的验证。

下表显示了具有特定选项的命令。"X" 表示命令包含该选项。

表 24-1 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"。通过加密来保护数据传输的保密性和完整性。但是，并非所有的 Kerberos 用户都可使用保密性服务。
- safe 将保护级别设置为 "safe"。通过加密校验和来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 455 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。

在下图中，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。

图 24-2 使用不可转发票证

1. (在 A 上) : `kinit david@ACME.ORG`



2. (在 A 上) : `kinit david@ACME.ORG`



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，您的配置可能会有所不同。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 要是可转发的，要么是不可转发的。

在下图中，用户 `david` 使用 `kinit -f` 获取可转发的 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。

图 24-3 使用可转发票证

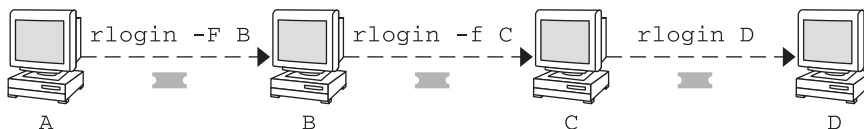
3. (在 A 上): `kinit -f david@ACME.ORG`



4. (在 A 上): `kinit -f david@ACME.ORG`



5. (在 A 上): `kinit -f david@ACME.ORG`



使用基于 Kerberos 的命令（示例）

以下示例说明基于 Kerberos 的命令选项的工作方式。

示例 24-5 将 `-a`、`-f` 和 `-x` 选项用于 `telnet`

在本示例中，用户 `david` 已经登录，并且要通过 `telnet` 远程登录到计算机 `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。

示例 24-5 将 -a、-f 和 -x 选项用于 telnet (续)

如果转发 Kerberos 票证，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 24-6 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 24-7 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
%
```

为了加密会话，joe 将保护级别设置为 private。

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 验证的工作原理的详细信息。

以下是本章中参考信息的列表。

- [第 461 页中的“Kerberos 文件”](#)
- [第 462 页中的“Kerberos 命令”](#)
- [第 463 页中的“Kerberos 守护进程”](#)
- [第 464 页中的“Kerberos 术语”](#)
- [第 468 页中的“Kerberos 验证系统的工作原理”](#)
- [第 469 页中的“使用 Kerberos 获取服务访问权限”](#)
- [第 472 页中的“使用 Kerberos 加密类型”](#)
- [第 473 页中的“使用 gsscred 表”](#)
- [第 474 页中的“Oracle Solaris Kerberos 和 MIT Kerberos 之间的显著差异”](#)

Kerberos 文件

本节列出了 Kerberos 服务所使用的一些文件。

表 25-1 Kerberos 文件

文件名	说明
~/gkadmin	用于在 SEAM Tool 中创建新主体的缺省值
~/k5login	向 Kerberos 帐户授予访问权限的主体的列表
/etc/krb5/kadm5.acl	Kerberos 访问控制列表文件，包括 KDC 管理员的主体名称及其 Kerberos 管理特权
/etc/krb5/kadm5.keytab	废弃：该文件已从 Oracle Solaris 11 发行版中移除。
/etc/krb5/kdc.conf	KDC 配置文件

表 25-1 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	网络应用服务器的密钥表文件
/etc/krb5/warn.conf	Kerberos 票证失效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 <i>uid</i> 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 <i>xxxxxx</i> 是一个随机字符串。
/var/krb5/.k5.REALM	KDC 存储文件，其中包含 KDC 主密钥的副本
/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.ulong	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 的备份文件
/var/krb5/slave_datatrans_slave	对指定的 <i>slave</i> 执行完全更新时创建的临时转储文件

Kerberos 命令

本节列出了 Kerberos 产品中包含的部分命令。

表 25-2 Kerberos 命令

命令	说明
/usr/bin/ftp	文件传输协议程序
/usr/bin/kdestroy	销毁 Kerberos 票证
/usr/bin/kinit	获取并缓存 Kerberos 票证授予票证
/usr/bin/klist	显示当前的 Kerberos 票证

表 25-2 Kerberos 命令 (续)

命令	说明
/usr/bin/kpasswd	更改 Kerberos 口令
/usr/bin/ktutil	管理 Kerberos 密钥表文件
/usr/bin/kvno	为 Kerberos 主体列出密钥版本号
/usr/bin/rcp	远程文件复制程序
/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/gsscred	管理 gsscred 表项
/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/kcclient	Kerberos 客户机安装脚本，有无安装配置文件皆可使用
/usr/sbin/kdb5_ldap_util	为 Kerberos 数据库创建 LDAP 容器
/usr/sbin/kdb5_util	创建 Kerberos 数据库和存储文件
/usr/sbin/kgcmgr	配置 Kerberos 主 KDC 和从 KDC
/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 25-3 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

表 25-3 Kerberos 守护进程 (续)

守护进程	说明
/usr/lib/krb5/krb5kdc	Kerberos 票证处理守护进程
/usr/lib/krb5/ktkt_warnd	Kerberos 票证失效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

以下章节介绍 Kerberos 的术语及其定义。整个 Kerberos 文档中使用了这些术语。要掌握 Kerberos 概念，必须先了解这些术语。

特定于 Kerberos 的术语

要管理 KDC，您需要了解本节中介绍的术语。

密钥分发中心 (Key Distribution Center, KDC) 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。

存储文件 包含 KDC 的主密钥。当重新引导服务器以在启动 `kadmind` 和 `krb5kdc` 命令之前自动验证 KDC 时，将使用此密钥。因为该文件包含主密钥，因此，该文件及其所有备份都应安全保存。该文件以 `root` 的只读权限创建。为确保文件安全，请勿更改其权限。如果该文件遭破坏，可以使用密钥访问或修改 KDC 数据库。

特定于验证的术语

要了解验证过程，您需要知道本节中介绍的术语。程序员和系统管理员应该熟悉这些术语。

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 软件将在此过程中发出许多请求。因此，区分该软件的操作和用户的操作非常重要。

术语 *server* 和 *service* 可以互换使用。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**对应于服务器上支持的特定功能（如 `ftp` 或 `nfs`）。文档通常会将服务器描述为服务的一部分，但此定义混淆了这些术语的含义。因此，术语 *server* 是指物理系统。术语 *service* 是指软件。

Kerberos 产品使用两种类型的密钥。一种类型是口令派生的密钥。口令派生的密钥分配给每个用户主体，且仅为用户和 KDC 所知。Kerberos 产品使用的另一种密钥类型是随机密钥，它与口令无关，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。

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：

- 服务的主体名称
- 用户的主体名称
- 用户主机的 IP 地址
- 时间戳
- 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 可颁发嵌入在如下所述的凭证中的票证。颁发后的票证可重复使用，直到过期。

凭证是一种信息包，包含票证和匹配的会话密钥。凭证使用请求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。

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、时间戳和其他数据。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。这些属性在创建票证时指定，但之后亦可修改。例如，可将票证从 `forwardable` 改为 `forwarded`。可以使用 `klist` 命令查看票证属性。请参见第 448 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张可转发票证，则前者可登录到自己的计算机，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，请参见示例 24-1。

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密钥。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（无效）

无效票证是一个尚不可用的以后生效票证。应用服务器会拒绝无效票证，直到其通过验证为止。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无法使用该票证。颁发以后生效的票证时，该票证尚未生效，且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截止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。Oracle 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截止时间：票证当前实例的截止时间，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，则可在第一个截止时间之前更新票证。例如，某个票证的有效期为一个小时，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），则该票证将自动过期且无法更新。

有关如何查看票证属性的信息，请参见第 448 页中的“查看 Kerberos 票证”。

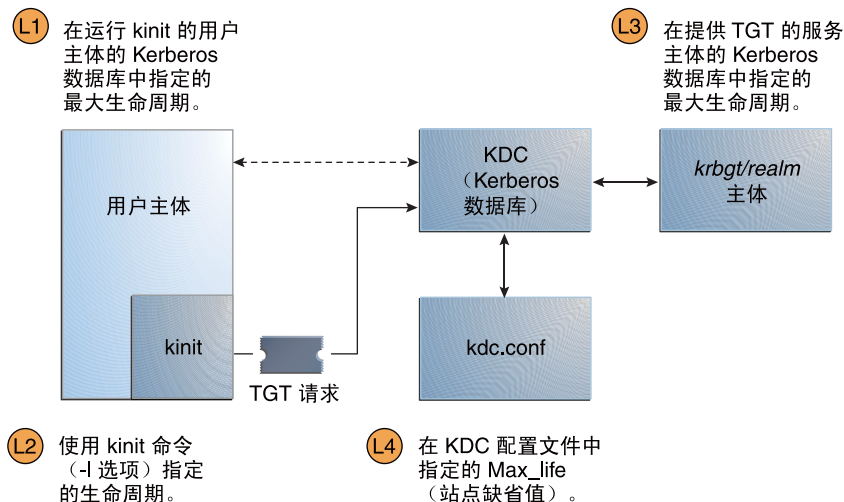
票证生命周期

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 25-1 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/realms` 主体）会提供最长生命周期值。

图 25-1 如何确定 TGT 的生命周期



票证生命周期 = L1、L2、L3 和 L4 中的最小值

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：

- 当使用 `kinit` 获取或更新票证时，通过 `kinit` 的 `-r` 选项指定的可更新生命周期值。
- `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/realms`。
- 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

每张票证都以主体名称标识。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 25-4 Kerberos 主体名称示例

Principal Name (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	kcliclient 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。
host/boston.example.com@EXAMPLE.COM	基于 Kerberos 的应用程序（如 klist 和 kprop）和服务（如 ftp 和 telnet）使用的主体。此主体称为 host 或服务主体，用于验证 NFS 挂载。客户机也会使用该主体验证 TGT 是否是从正确的 KDC 颁发给客户机的。
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秘密钥。每个主 KDC 都具有一个这样的主体。
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，用于接收来自未运行 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	与客户机上的 root 帐户关联的主体。该主体称为 root 主体，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 验证系统的工作原理

如果您可以提供证明身份的票证和匹配的会话密钥，应用程序将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即组成了凭证。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 的服务器上的 ftp 服务需要凭证。访问其他服务器中的 ftp 服务仍然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。

Kerberos 服务如何与 DNS 和 nsswitch 服务交互

Kerberos 服务编译为使用 DNS 解析主机名。完成主机名解析时，根本不会检查 nsswitch 服务。

使用 Kerberos 获取服务访问权限

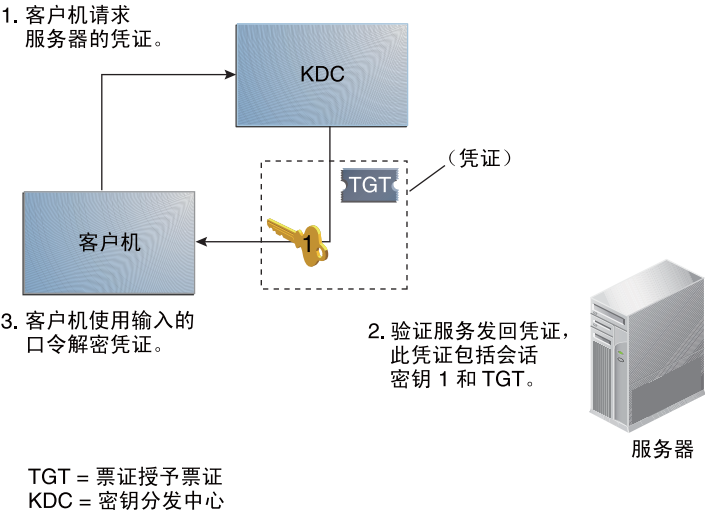
要访问特定服务器上的特定服务，用户必须获取两个凭证。第一个凭证是票证授予票证（即所知的 TGT）。票证授予服务解密此凭证后，该服务即可为用户请求访问的服务器创建第二个凭证。然后，可使用第二个凭证来请求访问该服务器中的相应服务。该服务器成功解密第二个凭证后，便会授予用户访问权限。以下各节详细介绍了此过程。

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，验证服务会获取该主体的私钥。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。

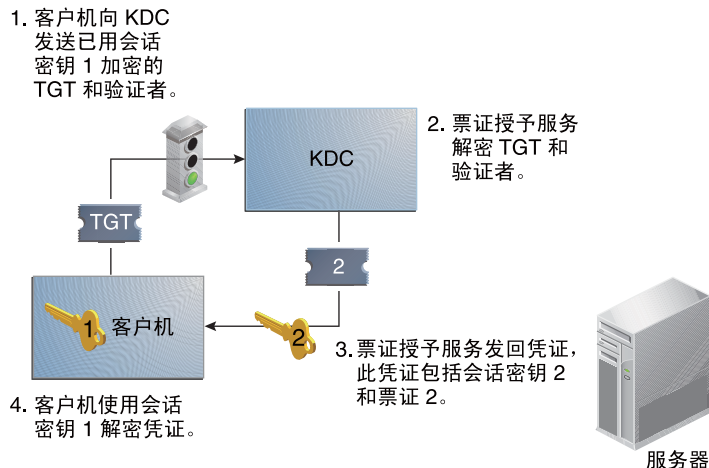
图 25-2 获取用于票证授予服务的凭证



获取服务器凭证

1. 要请求对特定服务器的访问权限，客户机首先必须向验证服务获取该服务器凭证。请参见第 469 页中的“获取用于票证授予服务的凭证”。客户机随后会向票证授予服务发送请求，其中包括服务主体名称、票证 1 和以会话密钥 1 加密的验证者。票证 1 最初是使用票证授予服务的服务密钥，由验证服务进行加密的。
2. 因为票证授予服务的服务密钥为票证授予服务所知，票证 1 可以解密。票证 1 中的信息包括会话密钥 1，这样，票证授予服务可以解密验证者。在此情况下，由票证授予服务对用户主体进行验证。
3. 一旦验证成功，票证授予服务会为用户主体和服务器（会话密钥 2）生成会话密钥，为服务器生成票证（票证 2）。然后，会使用会话密钥 1 对会话密钥 2 和票证 2 进行加密。因为会话密钥 1 仅为客户机和票证授予服务所知，因此该信息是安全的，且可以通过网络安全发送。
4. 客户机收到该信息包后，会使用会话密钥 1 解密信息，该密钥储存在凭证高速缓存中。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 25-3 获取服务器凭证

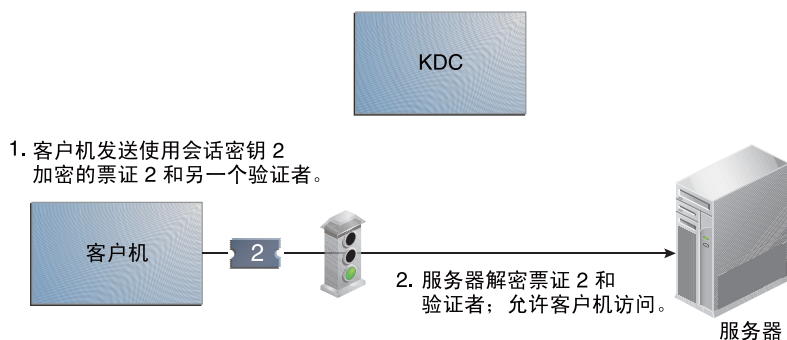


TGT = 票证授予票证
KDC = 密钥分发中心

获取对特定服务的访问权限

1. 要请求对特定服务的访问权限，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 469 页中的“获取用于票证授予服务的凭证”和第 470 页中的“获取服务器凭证”。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。验证者使用会话密钥 2 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 25-4 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 - 在 Solaris 10 8/07 之前的发行版中，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。

客户机向 KDC 请求票证时，KDC 必须使用其加密类型与客户机和服务器都兼容的密钥。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 - 如果安装的主 KDC 上未运行 Solaris 10 发行版，在升级主 KDC 之前，从 KDC 必须升级到 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。

下面列出了在更改加密类型前必须考虑的问题：

- 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。如果使用主体的系统不支持该缺省加密类型集，则您在创建主体时应限制这些加密类型。通过在 `kadmin addprinc` 中使用 `-e` 标志，或在 `kdc.conf` 文件中将 `supported_encetypes` 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_encetypes` 参数。如果设置 `supported_encetypes`，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_etype` 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_etype` 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_etype` 条目。另外，如果在 `kdc.conf` 中设置了 `supported_etypes`，则还应确保将 `master_key_etype` 设置为 `supported_etypes` 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_etypes` 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_etypes` 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_etypes` 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_etypes`，因为此参数可能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_etypes` 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_etypes` 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_etype` 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，则此参数可能会导致互操作性问题。
- 在服务器上，可使用 `kdc.conf` 中的 `permitted_etypes` 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 317 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，会检查 gsscred 表。

Oracle Solaris Kerberos 和 MIT Kerberos 之间的显著差异

Kerberos 服务的 Solaris 10 版基于 MIT Kerberos 1.2.1 版。下面列出了 Solaris 10 发行版中包括的 MIT 1.2.1 版中没有的增强功能：

- Oracle 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
- 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
- 使用加密框架进行加密

此版本还包括一些 MIT 1.2.1 后期错误修复。特别是，添加了 1.2.5 二叉树 (btree) 错误修复和 1.3 TCP 支持。

第 7 部分

在 Oracle Solaris 中审计

本部分提供有关审计子系统的配置、管理和使用的信息。

- 第 26 章，审计（概述）
- 第 27 章，规划审计
- 第 28 章，管理审计（任务）
- 第 29 章，审计（参考）

审计（概述）

Oracle Solaris 的审计子系统保留了有关如何使用系统的记录。审计服务包括帮助分析审计数据的工具。

本章介绍 Oracle Solaris 中审计的工作原理。以下是本章中信息的列表。

- 第 477 页中的“什么是审计？”
- 第 478 页中的“审计术语和概念”
- 第 485 页中的“审计如何与安全相关？”
- 第 485 页中的“审计工作原理”
- 第 486 页中的“配置审计的方式”
- 第 487 页中的“在具有 Oracle Solaris 区域的系统上审计”
- 第 488 页中的“关于此发行版中的审计服务”

有关规划建议，请参见第 27 章，[规划审计](#)。有关在站点上配置审计的过程，请参见第 28 章，[管理审计（任务）](#)。有关参考信息，请参见第 29 章，[审计（参考）](#)。

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，用户提供用户名并成功通过 PAM 验证后，将生成唯一且保持不变的**审计用户 ID**并与该用户相关联，还会生成唯一的审计会话 ID 并与该用户的进程相关联。在该登录会话期间启动的每个进程都会继承此审计会话 ID。从一个用户切换为另一个用户时，系统使用同一审计用户 ID 跟踪所有用户操作。有关切换身份的更多详细信息，请参见 [su\(1M\)](#) 手册页。请注意，缺省情况下，某些操作（例如引导和关闭系统）始终都需要进行审计。

使用审计服务可以：

- 监视主机上发生的与安全相关的事件
- 在网络范围的审计迹中记录事件

- 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对特权的扩展使用

审计术语和概念

以下术语用于说明审计服务。有些定义包含指向更完整说明的链接。

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 481 页中的“审计类和预选”以及 audit_flags(5)、audit_class(4) 和 audit_event(4) 手册页。
audit file system（审计文件系统）	二进制格式的审计文件系统信息库。 有关更多信息，请参见第 482 页中的“审计日志”和 audit.log(4) 手册页。
audit event（审计事件）	可审计的与安全相关的系统操作。为了便于选择，将事件分为多个审计类。 有关更多信息，请参见第 480 页中的“审计事件”和 audit_event(4) 手册页。
audit flag（审计标志）	作为参数提供给命令或关键字的审计类。标志的前缀可以是加号或减号，+ 表示对类的成功事件进行审计，而 - 表示对类的失败事件进行审计。在加号前面加上插入记号 ^ 表示将不审计成功事件 (^+)，而在减号前面加上该插入记号表示将不审计失败事件 (^-)。 有关更多信息，请参见 audit_flags(5) 手册页和第 557 页中的“审计类语法”。
audit plugin（审计插件）	用于将队列中的审计记录传输到指定位置的模块。audit_binfile 插件可创建二进制审计文件。而这些二进制文件组成了审计迹，存储在审计文件系统中。audit_remote 插件可将二进制审计记录发送到远程系统信息库。audit_syslog 插件可将选定的审计记录汇总到 syslog 日志中。

	有关更多信息，请参见第 482 页中的“审计插件模块”以及模块手册页 audit_binfile(5)、audit_remote(5) 和 audit_syslog(5)。
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，还包括是否在写满审计队列时暂停可审计的操作。 有关更多信息，请参见第 496 页中的“了解审计策略”和 auditconfig(1M) 手册页。
Audit record（审计记录）	审计队列中收集的审计数据。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。 有关更多信息，请参见第 481 页中的“审计记录和审计标记”和 audit.log(4) 手册页。
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。 有关更多信息，请参见第 561 页中的“审计标记格式”和 audit.log(4) 手册页。
audit trail（审计迹）	一个或多个审计文件的集合，用于存储使用缺省插件 audit_binfile 的所有已审计系统上的审计数据。 有关更多信息，请参见第 560 页中的“审计迹”。
post-selection（后选）	有关要在审计迹中检查哪些审计事件的选择。缺省活动插件 audit_binfile 可创建审计迹。后选工具 auditreduce 命令将从审计迹中选择记录。 有关更多信息，请参见 auditreduce(1M) 和 praudit(1M) 手册页。
preselection（预选）	有关要监视哪些审计类的选择。将在审计队列中收集预选的审计类的审计事件。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队列中。 有关更多信息，请参见第 481 页中的“审计类和预选”以及 audit_flags(5) 和 auditconfig(1M) 手册页。

public object（公共对象）

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。不会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

审计事件代表某系统上可审计的操作。审计事件在 /etc/security/audit_event 文件中列出。每个审计事件均连接到一个系统调用或用户命令，并指定给一个或多个审计类。有关 audit_event 文件的格式的说明，请参见 [audit_event\(4\)](#) 手册页。

例如，AUE_EXECVE 审计事件将审计 execve() 系统调用。命令 auditrecord -e execve 显示以下条目：

execve	
system call	execve
event ID	23
class	ps,ex
header	
path	
[attribute]	omitted on error
[exec_arguments]	output if argv policy is set
[exec_environment]	output if arge policy is set
subject	
[use_of_privilege]	
return	

如果预选审计类 ps 或 ex，则每个 execve() 系统调用都将记录在审计队列中。

审计过程处理**可归属**和**无归属**事件。审计策略将事件分为**同步**事件和**异步**事件，如下所示：

- **可归属事件**—可归属到某个用户的事件。execve() 系统调用可归属到某个用户，因此将该调用视为可归属事件。所有的可归属事件都是同步事件。
- **无归属事件**—在内核中断级别发生的事件，或在验证用户之前发生的事件。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。多数无归属事件都是异步事件。但是，具有关联进程的无归属事件（如登录失败）是同步事件。
- **同步事件**—与系统中的进程关联的事件。同步事件占系统事件的大多数。
- **异步事件**—与任何进程无关联的事件，因此既不会阻止进程，随后也不会唤醒进程。例如，初始系统引导和 PROM 进入和退出事件都是异步事件。

除了审计服务定义的审计事件，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。供应商需要联系其 Oracle Solaris 代表，以保留事件编号并获取对审计界面的访问权限。

审计类和预选

每个审计事件属于一个或多个**审计类**。审计类是用于容纳大量审计事件的方便容器。**预选**要审计的类时，该类中的所有事件都将记录在审计队列中。例如，预选 `ps` 审计类时，将记录 `execve()`、`fork()` 以及其他系统调用。

可以预选系统中的事件和特定用户启动的事件。

- **系统范围的预选**—通过在 `auditconfig` 命令中使用 `-setflags` 和 `-setnaflags` 选项指定系统范围内的审计缺省值。

注—如果已设置 `perzone` 策略，则可以在每个区域中指定缺省审计类。对于 `perzone` 审计，缺省值为区域范围，而非系统范围。

- **特定于用户的预选**—通过为用户配置审计标志，指定各个用户的审计值与系统范围的审计缺省值之间的差异。`useradd`、`roleadd`、`usermod` 以及 `rolemod` 命令将 `audit_flags` 安全属性放置在 `user_attr` 数据库中。`profiles` 命令将权限配置文件的审计标志放置在 `prof_attr` 数据库中。

审计预选掩码确定要针对用户审计的事件类。有关用户预选掩码的说明，请参见第 559 页中的“[进程审计特征](#)”。要了解使用哪些已配置审计标志，请参见第 187 页中的“[搜索指定安全属性的顺序](#)”。

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，`lo` 和 `ps` 类定义在 `audit_class` 文件中显示为：

```
0x0000000000001000:lo:login or logout
0x00000000000100000:ps:process start/stop
```

审计类包括以下两个全局类：`all` 和 `no`。[audit_class\(4\)](#) 手册页中介绍了这些审计类。有关类的列表，请阅读 `/etc/security/audit_class` 文件。

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，还可以创建新类以包含选定事件。有关过程，请参见第 514 页中的“[如何更改审计事件的类成员身份](#)”。要查看映射到类的事件，请使用 `auditrecord -c class` 命令。

审计记录和审计标记

每条**审计记录**记录一个发生的审计事件。该记录包含操作执行者、受影响的文件、尝试执行的操作以及操作发生的时间和位置等信息。下面是一条 `login` 审计记录的示例：

```
header,69,2,login - local,,example_system,2010-10-10 10:10:10.020 -07:00
subject,jdoe,jdoe,staff,jdoe,staff,1210,4076076536,69 2 example_system
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，记录中都会包含为该事件定义的部分或全部标记。事件的性质决定了要记录的标记。在前面的示例中，每行都以审计标记的名称开头。标记名称后跟审计标记的内容。`header`、`subject` 和 `return` 审计标记一起组成了 `login - local` 审计记录。要显示组成审计记录的标记，请使用 `auditrecord -e event` 命令。

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 561 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。

审计插件模块

可以指定哪些审计插件模块处理您的预选已经放入审计队列中的记录。至少有一个插件必须处于活动状态。缺省情况下，`audit_binfile` 插件处于活动状态。可使用 `auditconfig -setplugin plugin-name` 命令配置插件。

审计服务提供以下插件：

- `audit_binfile` 插件—处理将审计队列发送到二进制审计文件的过程。有关更多信息，请参见 [audit.log\(4\)](#) 手册页。
- `audit_remote` 插件—处理将二进制审计记录从审计队列安全发送到已配置的远程服务器的过程。`audit_remote` 插件使用 `libgss()` 库验证服务器。出于保密性和完整性的考虑，传输过程受到保护。
- `audit_syslog` 插件—处理将选定记录从审计队列发送到 `syslog` 日志的过程。

要配置插件，请参见 [auditconfig\(1M\)](#) 手册页。有关插件配置的示例，请参见第 515 页中的“[配置审计日志（任务）](#)”中的任务。

有关插件的信息，请参见 [audit_binfile\(5\)](#)、[audit_remote\(5\)](#) 和 [audit_syslog\(5\)](#) 手册页。

审计日志

审计记录是在审计日志中收集的。对于审计记录，审计服务提供三种输出模式。

- 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集提供完整的审计记录。完整的审计记录称为**审计迹**。这些日志由 `audit_binfile` 插件创建，可通过 `praudit` 和 `auditreduce` 后选命令查看。
- `audit_remote` 插件可将审计记录流化处理到远程系统信息库。该系统信息库负责维护审计迹并提供后选工具。
- `syslog` 实用程序收集并存储审计记录的文本摘要。`syslog` 记录不是完整的记录。以下示例显示了 `login` 审计记录的 `syslog` 项：

```
Oct 10 10:10:20 example_system auditd: [ID 6472 audit.notice] \
login - login ok session 4076172534 by root as root:other
```

站点可以配置审计以收集所有格式的审计记录。可以在您的站点上配置系统，使其在本地使用二进制模式，将二进制文件发送到远程系统信息库，使用 `syslog` 模式，或使用以上这些模式的任意组合。下表对二进制审计记录和 `syslog` 审计记录进行了比较。

表 26-1 二进制审计记录、远程审计记录以及 `syslog` 审计记录的比较

功能	二进制和远程记录	<code>syslog</code> 记录
协议	二进制—写入文件系统 远程—流化处理到远程系统信息库	将 UDP 用于远程日志记录
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	二进制—存储在系统上的 <code>zpool</code> 中 远程—远程系统信息库	存储在 <code>syslog.conf</code> 文件中指定的位置
配置方式	二进制—在 <code>audit_binfile</code> 插件上设置 <code>p_dir</code> 属性 远程—在 <code>audit_remote</code> 插件上设置 <code>p_hosts</code> 属性，并使该插件处于活动状态	使 <code>audit_syslog</code> 插件处于活动状态并配置 <code>syslog.conf</code> 文件
读取方式	二进制—在批处理模式下，浏览器通常以 XML 格式输出 远程—系统信息库指定相应过程	实时读取，或者通过为 <code>syslog</code> 创建脚本来进行搜索 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
时间戳	国际协调时间 (Coordinated Universal Time, UTC)	审计的系统时间

二进制记录提供最高的安全性和最大的覆盖范围。二进制输出满足安全证书的要求，例如通用准则 (<http://www.commoncriteriaportal.org/>) 审计要求。

`audit_binfile` 插件可将记录写入对其进行保护以防止偷窥的文件系统中。在单个系统上，将收集所有二进制记录并按顺序显示它们。当某个审计迹内的系统分布于不同的时区时，可以参考二进制日志中的 UTC 时间戳进行精确比较。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。

`audit_remote` 插件可将记录写入远程系统信息库。该系统信息库负责存储和后选。

相反，`syslog` 记录可能会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戳的审计记录摘要。您可以使用为来自各种源（包括工作站、服务器、防火墙和路由器）的 `syslog` 消息开发的同一管理和分析工具。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` 项限制为 1024 个字符，因此可能会截断日志中的某些审计记录。在单个系统上，不会收集所有的审计记录。记录可能不会按顺序显示。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，因此不能根据时间戳为多个系统构造审计迹。

有关插件和审计日志的更多信息，请参阅以下内容：

- [audit_binfile\(5\)](#) 手册页
- [audit_syslog\(5\)](#) 手册页
- [audit.log\(4\)](#) 手册页
- 第 518 页中的“如何为审计迹指定审计空间”
- 第 521 页中的“如何配置 `syslog` 审计日志”

存储和管理审计迹

`audit_binfile` 插件处于活动状态时，**审计文件系统** 以二进制格式保留审计文件。典型的安装使用 `/var/audit` 文件系统，也可以使用其他文件系统。所有审计文件系统的内容组成了**审计迹**。审计记录按以下顺序存储在这些文件系统中：

- **主审计文件系统**—`/var/audit` 文件系统，即用于系统审计文件的缺省文件系统
- **辅助审计文件系统**—按管理员意愿在其中放置系统审计文件的文件系统

这些文件系统作为 `audit_binfile` 插件的 `p_dir` 属性参数来指定。对于某个文件系统，列表中位于此文件系统前面的文件系统已满时才会使用此文件系统。有关包含文件系统项列表的示例，请参见第 515 页中的“如何为审计文件创建 ZFS 文件系统”。

将审计文件放置在缺省审计根目录下可帮助审计审阅者审阅审计迹。`auditreduce` 命令使用审计根目录来查找审计迹中的所有文件。缺省审计根目录为 `/var/audit`。可以在 `auditreduce` 命令中使用 `-M` 选项以指定特定计算机中的审计文件，而使用 `-S` 选项可以指定其他审计文件系统。有关更多信息，请参见 [auditreduce\(1M\)](#) 手册页。

审计服务提供用于合并和过滤审计迹文件的命令。`auditreduce` 命令可以合并审计迹中的审计文件。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。

确保时间戳可靠

合并多个系统中的审计日志时，这些系统上的日期和时间必须准确。同样，将审计日志发送到远程系统时，记录系统和系统信息库系统必须具有准确的时钟。网络时间协

议 (Network Time Protocol, NTP) 可以使各个系统时钟保持准确和相互协调。有关更多信息，请参见《[Oracle Solaris 管理：网络服务](#)》中的第 3 章“与时间有关的服务”和 [xntpd\(1M\)](#) 手册页。

管理远程系统信息库

`audit_remote` 插件处于活动状态时，远程系统信息库将管理审计记录。

审计如何与安全相关？

审计通过显示可疑或异常的系统使用模式来帮助检测潜在的安全违规。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。知道自己的活动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。

最佳做法要求在用户登录或开始系统进程之前设置审计参数，因为大部分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。第 27 章，[规划审计](#)和第 28 章，[管理审计（任务）](#)中详细介绍了审计服务如何监视和报告这些事件。

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

审计工作原理

审计在发生指定的事件时生成审计记录。通常情况下，生成审计记录的事件包括：

- 系统启动和系统关闭
- 登录和注销
- 进程创建或进程销毁，或线程创建或线程销毁
- 打开、关闭、创建、销毁或重命名对象
- 使用特权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
- `asynchronous audit event`（异步审计事件）的结果
- 进程系统调用的结果

捕获相关的事件信息后，会将这些信息格式化为审计记录。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。该记录随后会被放置在活动**插件**的审计队列中。尽管所有插件都可以处于活动状态，但至少有一个插件必须处于活动状态。

缺省情况下，有一个插件处于活动状态。这个插件即为 `audit_binfile` 插件，它将审计记录写入审计文件。这些文件以二进制格式存储在本地。活动的 `audit_remote` 插件可将这些记录发送到远程系统信息库。活动的 `audit_syslog` 插件可将文本摘要发送到 `syslog` 实用程序。有关图示，请参见图 26-1。

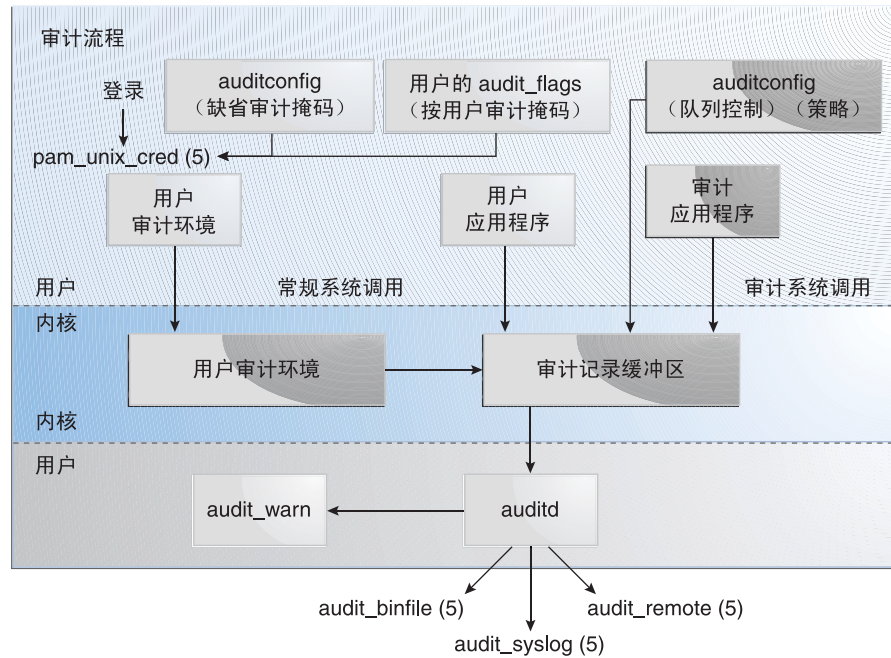
本地存储后，审计文件可以位于一个或多个 ZFS 池中。ZFS 池可使本地存储易于管理。这些池可以位于不同的系统中，也可以位于不同但链接的网络中。相互链接的审计文件集合称为**审计迹**。

有关更多信息，请参见第 486 页中的“配置审计的方式”、第 482 页中的“审计日志”和第 482 页中的“审计插件模块”。

配置审计的方式

在系统配置期间，可以**预选**要监视的审计记录的类。还可以针对单个用户微调执行审计的程度。下图显示了 Oracle Solaris 审计流程的详细信息。

图 26-1 审计流程



在内核中收集审计数据后，插件会将数据分发到相应位置。

- `audit_binfile` 插件可将二进制审计记录放置在 `/var/audit` 文件系统中。通过后选工具，可以检查审计迹中感兴趣的部分。
- `audit_remote` 插件可将受保护链接中的二进制审计记录发送到远程系统信息库。
- `audit_syslog` 插件可将审计记录的文本摘要发送到 `syslog` 实用程序。

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 556 页中的“[审计和 Oracle Solaris 区域](#)”。

在具有 Oracle Solaris 区域的系统上审计

区域 (zone) 是在 Oracle Solaris OS 单一实例中创建的虚拟化操作系统环境。审计服务审计整个系统，包括区域中的活动。安装了非全局区域的系统可以运行单一审计服务，以相同方式审计所有区域。它还可为每个区域（包括全局区域）运行一个审计服务。

满足以下条件的站点可以运行单一审计服务：

- 站点需要单映像审计迹。
- 非全局区域用作应用程序容器。区域是一个管理域的组成部分。也就是说，非全局区域都没有定制的命名服务文件。

如果系统上的所有区域都在一个管理域中，可以使用 `zonename` 审计策略区分在不同区域配置的审计事件。

- 管理员希望降低审计开销。全局区域管理员以同样的方式审计所有区域。而且，全局区域的审计守护进程为系统上的所有区域服务。

满足以下条件的站点可为每个区域运行一个审计服务：

- 站点不需要单映像审计迹。
- 非全局区域有定制的命名服务文件。这些独立的管理域通常充当服务器。
- 各区域管理员希望控制他们管理的区域中的审计。在按区域审计中，区域管理员可以决定是在他们管理的区域中启用还是禁用审计。

按区域审计的优点在于，可为每个区域定制审计迹，并且能够按区域禁用区域审计。但是，这些优点可能会被管理开销抵销。每个区域管理员都必须管理审计。每个区域运行自己的审计守护进程，有自己的审计队列和审计日志。必须管理这些审计日志。

关于此发行版中的审计服务

审计具备以下特性：

- 审计即是服务。请参见第 553 页中的“审计服务”。
- 缺省情况下启用审计。
- 禁用或启用审计服务时，无需进行重新引导。
- `auditconfig` 命令用于显示和更改审计策略、无归属标志、可归属标志、插件以及队列控件。请参见 `auditconfig(1M)` 手册页。
- 公共对象的审计在审计迹中生成的噪音较少。
- 非内核事件的审计不产生任何性能影响。
- 缺省情况下，`login/logout` 类中的事件会针对系统和 `root` 帐户进行审计。
- Oracle Solaris 提供三个插件：`audit_binfile`、`audit_remote` 和 `audit_syslog`。请参见 `audit_binfile(5)`、`audit_remote(5)` 和 `audit_syslog(5)` 手册页。
- 非全局区域可以在不必审计全局区域的情况下进行审计。在非全局区域中进行审计的唯一要求是在全局区域中设置 `perzone` 审计策略。
- 审计类的可能数目从 32 增加到 64。前八个高级位专为客户预留。
- 审计的权限配置文件已重新配置。请参见第 555 页中的“用于管理审计的权限配置文件”。

- `audit_flags` 安全属性用于配置与系统范围的审计的用户差异。该关键字是 `useradd`、`usermod`、`roleadd` 和 `rolemod` 命令的一个参数。`audit_flags` 值存储在 `user_attr` 数据库中。请参见 [useradd\(1M\)](#)、[usermod\(1M\)](#)、[roleadd\(1M\)](#)、[rolemod\(1M\)](#) 和 [user_attr\(4\)](#) 手册页。
`profiles` 命令的关键字 `always_audit` 和 `never_audit` 可以更新 `prof_attr` 数据库中的 `audit_flags` 安全属性。有关更多信息，请参见 [profiles\(1\)](#) 手册页和 [第 187 页](#) 中的“搜索指定安全属性的顺序”。
- 定义了新的审计类。`ft` 审计类包含文件传输审计事件。`ftp` 和 `sftp` 命令包含在由该类审计的事件中。`frcp` 审计类包含记录的审计事件，无论管理员是否预选了这些审计事件。`auditrecord -c classname` 命令描述了这些新类中的审计事件。

规划审计

本章介绍如何针对 Oracle Solaris 安装定制审计服务。以下是本章介绍的规划信息的列表：

- [第 491 页](#)中的“规划审计（任务）”
- [第 496 页](#)中的“了解审计策略”
- [第 498 页](#)中的“控制审计成本”
- [第 499 页](#)中的“有效审计”

有关审计的概述，请参见[第 26 章，审计（概述）](#)。有关在站点上配置审计的过程，请参见[第 28 章，管理审计（任务）](#)。有关参考信息，请参见[第 29 章，审计（参考）](#)。

规划审计（任务）

您需要认真选择要审计的活动类型，同时还需要收集有用的审计信息。您还需要仔细规划要审计的对象及内容。如果使用缺省 `audit_binfile` 插件，审计文件会快速增长进而填满可用空间，因此必须分配足够的磁盘空间。

以下任务列表列出了规划磁盘空间以及要记录的事件时所需执行的主要任务。

任务	参考
确定针对非全局区域的审计策略	第 492 页 中的“如何规划区域中的审计”
规划审计迹的存储空间	第 493 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 494 页 中的“如何规划要审计的对象及内容”

▼ 如何规划区域中的审计

如果系统包含非全局区域，可通过审计全局区域来审计这些区域，或者可以单独为每个非全局区域配置、启用和禁用审计服务。例如，您可以仅审计非全局区域，而不审计全局区域。

有关如何权衡选择的介绍，请参见第 487 页中的“在具有 Oracle Solaris 区域的系统上审计”。

- 选择以下选项之一：

- 方法 1 – 为所有区域配置单个审计服务。

以相同方式审计所有区域可以创建一个单映像审计迹。使用 `audit_binfile` 或 `audit_remote` 插件且系统上的所有区域都属于一个管理域时，会出现单映像审计迹。然后，可以很容易地比较审计记录，因为每个区域的记录都以相同的设置预选。

该配置将所有区域视为一个系统的组成部分。全局区域在一个系统上只运行一个审计服务，并收集所有区域的审计记录。您只能在全局区域中定制 `audit_class` 和 `audit_event` 文件，然后将这些文件复制到各个非全局区域。

- a. 对每个区域使用相同的命名服务。

注 – 如果已在非全局区域中定制命名服务文件，但未设置 `perzone` 策略，那么需要谨慎地使用审计工具来选择可用的记录。一个区域中的用户 ID 可以指不同区域中使用相同 ID 的不同用户。

- b. 启用包括区域名称在内的审计记录。

要使区域名称成为审计记录的一部分，请在全局区域中设置 `zonename` 策略。然后，可以使用 `auditreduce` 命令按区域从审计迹中选择审计事件。有关示例，请参见 `auditreduce(1M)` 手册页。

要规划单映像审计迹，请参阅第 494 页中的“如何规划要审计的对象及内容”。从第一步开始。全局区域管理员还必须留出部分存储，如第 493 页中的“如何规划审计记录的存储”中所述。

- 方法 2 – 为每个区域配置一个审计服务。

如果不同区域使用不同的命名服务数据库，或者区域管理员想要控制他们区域中的审计，请选择配置按区域审计。

注 – 要审计非全局区域，必须设置 `perzone` 策略，但不必在全局区域中启用审计服务。配置非全局区域审计，并从全局区域单独启用和禁用其审计服务。

- 配置按区域审计时，需要在全局区域中设置 `perzone` 审计策略。如果在第一次引导非全局区域之前设置按区域审计，则在第一次引导区域时开始审计。要设置审计策略，请参见第 525 页中的“如何配置每区域审计”。
- 每个区域管理员为该区域配置审计。
非全局区域管理员可以设置 `perzone` 和 `ahlt` 以外的全部策略选项。
- 每个区域管理员都可以启用或禁用区域中的审计。
- 要在审查期间生成可追溯到其来源区域的记录，请设置 `zonename` 审计策略。

要规划按区域审计，请参见第 494 页中的“如何规划要审计的对象及内容”。可以跳过第一步。如果 `audit_binfile` 插件处于活动状态，每个区域管理员还必须为每个区域留出部分存储，如第 493 页中的“如何规划审计记录的存储”中所述。

▼ 如何规划审计记录的存储

`audit_binfile` 插件可创建审计迹。审计迹需要专用的文件空间。此空间必须可用且安全。对于初始存储，系统使用 `/var/audit` 文件系统。您可以为审计文件配置其他审计文件系统。以下过程介绍了规划审计迹存储时必须解决的问题。

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 492 页中的“如何规划区域中的审计”。

您正在使用 `audit_binfile` 插件。

1 确定站点所需的审计量。

针对审计迹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 498 页中的“控制审计成本”和第 499 页中的“有效审计”。

有关实际步骤，请参见第 542 页中的“如何减少生成的审计记录数目”、第 549 页中的“如何压缩专用文件系统上的审计文件”和示例 28–28。

2 确定要审计的系统并配置它们的审计文件系统。

创建计划使用的所有文件系统的列表。有关配置准则，请参见第 484 页中的“存储和管理审计迹”和 `auditreduce(1M)` 手册页。要指定审计文件系统，请参见第 518 页中的“如何为审计迹指定审计空间”。

3 同步所有系统上的时钟。

有关更多信息，请参见第 484 页中的“确保时间戳可靠”。

▼ 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查看第 492 页中的“如何规划区域中的审计”。

1 确定是否需要单系统映像审计迹。

注 - 此步骤仅适用于 audit_binfile 插件。

单一管理域中的系统可以创建单系统映像审计迹。如果您的系统使用不同的命名服务，请从步骤 2 开始。然后，针对每个系统完成其余的规划步骤。

要为站点创建单系统映像审计迹，安装中的每个系统都应进行如下配置：

- 针对所有系统，使用相同的命名服务。
要正确解释审计记录，passwd、group 和 hosts 文件必须一致。
- 在所有系统上以相同的方式配置审计服务。有关显示和修改服务设置的信息，请参见 auditconfig(1M) 手册页。
- 对于所有系统，使用相同的 audit_warn、audit_event 和 audit_class 文件。

2 确定审计策略。

缺省情况下，仅启用 cnt 策略。

使用 auditconfig -lspolicy 命令查看可用策略选项的说明。

- 有关策略选项的影响，请参见第 496 页中的“了解审计策略”。
- 有关 cnt 策略的影响，请参见第 558 页中的“同步事件和异步事件的审计策略”。
- 要设置审计策略，请参见第 508 页中的“如何更改审计策略”。

3 确定是否要修改事件到类的映射。

几乎所有情况下，缺省映射便已够用。但是，如果添加新类、更改类定义或确定特定系统调用的记录没有用处，可能需要修改事件到类的映射。

有关示例，请参见第 514 页中的“如何更改审计事件的类成员身份”。

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间是在用户登录到系统之前。

在 auditconfig 命令中使用 -setflags 和 -setnaflags 选项预选的审计类适用于所有用户和进程。可以针对成功、失败或两者预选类。

有关审计类的列表，请阅读 /etc/security/audit_class 文件。

5 确定系统范围预选的用户修改。

如果您确定某些用户应该按照与系统不同的方式进行审计，请使用 `useradd`、`usermod`、`roleadd` 或 `rolemod` 命令的 `audit_flags` 安全属性。您还可以使用 `profiles` 命令将该属性添加到 `prof_attr` 数据库中的权限配置文件。为使用带有显式审计标志的权限配置文件的用户修改用户预选掩码。

有关过程，请参见第 505 页中的“如何配置用户审计特征”。有关有效的审计标志值，请参见第 187 页中的“搜索指定安全属性的顺序”。

6 决定如何管理 `audit_warn` 电子邮件别名。

每当审计系统检测到需要管理干预的情况时，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 512 页中的“如何配置 `audit_warn` 电子邮件别名”。

7 确定收集审计记录的格式和位置。

您有三个选择。

- 缺省情况下，在本地存储二进制审计记录。缺省存储目录为 `/var/audit`。要进一步配置 `audit_binfile` 插件，请参见第 515 页中的“如何为审计文件创建 ZFS 文件系统”。
 - 使用 `audit_remote` 插件将二进制审计记录流化处理到远程受保护的系统信息库。您必须指定文件的接收者。有关过程，请参见第 520 页中的“如何向远程系统信息库发送审计文件”。
 - 使用 `audit_syslog` 插件将审计记录摘要发送到 `syslog`。有关过程，请参见第 521 页中的“如何配置 `syslog` 审计日志”。
- 有关二进制和 `syslog` 格式的比较，请参见第 482 页中的“审计日志”。

8 确定向管理员发出有关收缩磁盘空间的警告的时间。

注 - 此步骤仅适用于 `audit_binfile` 插件。

当审计文件系统上的磁盘空间低于最低空闲空间百分比或软限制时，审计服务将会切换到下一个可用审计目录。然后，服务将发送一条警告，指出已超过软限制。

要设置最低空闲空间百分比，请参见示例 28-17。

9 决定当所有审计目录已满时需要执行的操作。

注 - 此步骤仅适用于 `audit_binfile` 插件。

在缺省配置中，`audit_binfile` 插件处于活动状态，并且设置了 `cnt` 策略。在此配置中，内核审计队列已满时，系统将继续工作。系统会对丢弃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，然后启用 `ahlt` 策略。当异步事件无法放入审计队列时，`ahlt` 策略会停止系统。

有关这些策略选项的说明，请参见第 558 页中的“同步事件和异步事件的审计策略”。要配置这些策略选项，请参见示例 28–6。

但是，如果 `audit_binfile` 队列已满，而另一活动插件的队列未满，则内核队列将继续向未满的插件发送记录。`audit_binfile` 队列可再次接收记录时，审计服务将恢复向其发送记录。

注 – 如果至少一个插件的队列正在接收审计记录，则不会触发 `cnt` 或 `ahlt` 策略。

了解审计策略

审计策略确定本地系统审计记录的特征。可以使用 `auditconfig` 命令设置这些策略。有关更多信息，请参见 `auditconfig(1M)` 手册页。

缺省情况下，会禁用大多数审计策略选项以最大程度地减少存储需求和系统处理需求。这些选项是审计服务的属性，并确定在系统引导时生效的策略。有关更多信息，请参见 `auditconfig(1M)` 手册页。

参考下表确定启用一个或多个审计策略选项而造成的额外开销是否与站点的需求相适应。

表 27–1 审计策略选项的影响

Policy Name (策略名称)	说明	更改策略选项的原因
ahlt	该策略仅适用于异步事件。禁用后，该策略允许在不生成审计记录的情况下完成事件。 启用后，该策略会在审计队列已满时停止系统。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该策略。该策略影响所有区域。	当系统可用性比安全性更重要时，适合禁用该选项。 在安全性极为重要的环境中，适合启用该选项。有关更全面的介绍，请参见第 558 页中的“同步事件和异步事件的审计策略”。
arge	禁用后，该策略将省略 <code>execve</code> 审计记录中已执行程序的环境变量。 启用后，该策略会将已执行程序的环境变量添加到 <code>execve</code> 审计记录。与禁用该策略的情况相比，生成的审计记录包含的详细信息要多得多。	与启用该选项的情况相比，禁用该选项后收集的信息要少很多。有关比较，请参见第 544 页中的“如何审计用户执行的所有命令”。 审计许多用户时，适合启用该选项。怀疑 <code>ex</code> 审计类的程序中使用的环境变量有问题时，也可以使用此选项。

表 27-1 审计策略选项的影响 (续)

Policy Name (策略名称)	说明	更改策略选项的原因
argv	禁用后, 该策略将省略 <code>execve</code> 审计记录中已执行程序参数。 启用后, 该策略会将已执行程序参数添加到 <code>execve</code> 审计记录。与禁用该策略的情况相比, 生成的审计记录包含的详细信息要多得多。	与启用该选项的情况相比, 禁用该选项后收集的信息要少很多。有关比较, 请参见第 544 页中的“如何审计用户执行的所有命令”。 审计许多用户时, 适合启用该选项。如果您有理由确信 <code>ex</code> 审计类中的异常程序正在运行, 也可以使用此选项。
cnt	禁用后, 该策略将阻止用户或应用程序运行。由于审计队列已满而导致审计记录无法添加到审计迹时, 会发生阻止。 启用后, 该策略允许在不生成审计记录的情况下完成事件。该策略将保留丢弃的审计记录数。	在安全性极为重要的环境中, 适合禁用该选项。 当系统可用性比安全性更重要时, 适合启用该选项。有关更全面的介绍, 请参见第 558 页中的“同步事件和异步事件的审计策略”。
group	禁用后, 该策略不会在审计记录中添加组列表。 启用后, 该策略会在每条审计记录中添加组列表作为特殊标记。	禁用该选项通常可以满足站点的安全要求。 当您需要审计主题所属的补充组时, 启用该选项很有用。
path	禁用后, 该策略会在每条审计记录中最多记录一条在系统调用期间使用的路径。 启用后, 该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。	禁用该选项最多在审计记录中放置一条路径。 启用该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 <code>path</code> 标记。
perzone	禁用后, 该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服务。如果预选 <code>zonename</code> 审计标记, 可在审计记录中找到特定区域中的审计事件。 启用后, 该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。审计服务运行在各个区域中。只能在全局区域中启用该策略。	当您没有特殊理由为每个区域维护单独的审计日志、队列和守护进程时, 禁用该选项很有用。 当仅通过检查带有 <code>zonename</code> 审计标记的审计记录无法有效监视系统时, 启用该选项很有用。
public	禁用后, 如果预选了文件读取, 则该策略不会将公共对象的只读事件添加到审计迹。包含只读事件的审计类包括 <code>fr</code>、<code>fa</code> 和 <code>cl</code>。 启用后, 如果预选了适当的审计类, 则该策略会记录公共对象的每个只读审计事件。	禁用该选项通常可以满足站点的安全要求。 启用该选项的作用很小。
seq	禁用后, 该策略不会将序列号添加到每条审计记录中。 启用后, 该策略会将序列号添加到每条审计记录中。<code>sequence</code> 标记保留序列号。	当审计顺利进行时, 禁用该选项便足以满足要求。 启用 <code>cnt</code> 策略后, 适合启用该选项。使用 <code>seq</code> 策略可以确定废弃数据的时间。或者, 可以使用 <code>auditstat</code> 命令查看丢弃的记录。

表 27-1 审计策略选项的影响 (续)

Policy Name (策略名称)	说明	更改策略选项的原因
trail	禁用后，该策略不会将 trailer 标记添加到审计记录中。 启用后，该策略会将 trailer 标记添加到每条审计记录中。	禁用该选项会创建一条较小的审计记录。 启用该选项会使用 trailer 标记清楚地标记每条审计记录的末尾。trailer 标记经常与 sequence 标记结合使用。使用 trailer 标记有助于恢复损坏的审计迹。
zonename	禁用后，该策略不会在审计记录中包括 zonename 标记。 启用后，该策略会在每条审计记录中包括 zonename 标记。	无需跟踪各个区域中的审计行为时，禁用该选项很有用。 要通过根据区域后选记录来分离和比较各个区域中的审计行为时，启用该选项很有用。

控制审计成本

由于审计会消耗系统资源，因此必须控制记录的详细程度。决定要审计的内容时，请考虑以下审计成本：

- 延长处理时间产生的成本
- 分析审计数据产生的成本

如果使用缺省插件 `audit_binfile`，您还必须考虑审计数据的存储成本。

延长审计数据处理时间产生的成本

延长处理时间产生的成本是审计成本中最不重要的部分。第一个原因是，在执行计算密集型任务（如图像处理、复杂计算等）时一般不会进行审计。如果使用 `audit_binfile` 插件，另一个原因是，审计管理员可以将后选任务从受审计系统移动到专用于分析审计数据的系统。最后，如果不预选内核事件，那么除了审计服务影响之外，内核对系统性能没有明显的影响。

分析审计数据产生的成本

分析成本大致上与收集的审计数据量成正比。分析成本包括合并与查看审计记录所需的时间，

对于 `audit_binfile` 插件所收集的记录，成本还包括归档记录及其支持名称服务数据库以及妥善保存记录所需的时间。支持数据库包括 `groups`、`hosts` 和 `passwd`。

生成的记录越少，分析审计迹所需的时间就越少。第 499 页中的“存储审计数据产生的成本”和第 499 页中的“有效审计”部分介绍了如何高效地执行审计。有效的审计可以减少审计数据量，同时提供足够的覆盖范围以实现站点的安全目标。

存储审计数据产生的成本

如果使用 `audit_binfile` 插件，存储成本是最主要的审计成本。审计数据量取决于以下各项：

- 用户数
- 系统数
- 使用量
- 所需的可追溯与可定责程度

由于上述因素随站点不同而不同，因此没有公式可以预先确定为审计数据存储预留的磁盘空间量。可以使用下列信息作为参考：

- 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。
- 精心预选审计类，以减少生成的记录量。
完全审计（即，使用 `all` 类）会很快填满磁盘空间。即使是简单的任务（例如编译某个程序）也可能会生成很大的审计文件。一个大小中等的程序在一分钟之内就可能生成数以千计的审计记录。
例如，省略 `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，有时也可以减少审计量。例如，与针对所有 `file_read` 事件进行审计相比，针对失败的 `file_read` 操作进行审计（即，使用 `-fr`）而生成的记录会少很多。
- 如果使用 `audit_binfile` 插件，有效的审计文件管理也很重要。例如，您可以压缩一个专用于审计文件的 ZFS 文件系统。
- 确立站点审计的思路。
将您的思路建立在合理的度量之上。这些度量包括站点所需的可追溯程度，以及管理的用户类型。

有效审计

以下方法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。

- 对于尽可能多的审计类，仅针对用户和角色而不是系统范围预选这些类。
- 一次只能对特定百分比的用户进行随机审计。
- 如果 `audit_binfile` 插件处于活动状态，可通过过滤、合并以及压缩文件来减少审计文件的磁盘存储要求。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。
- 实时监控审计数据有无异常行为。
 - `audit_syslog` 插件—您可以扩展已开发的管理和分析工具，以处理 `syslog` 文件中的审计记录。

- `audit_binfile` 插件—您可以设置针对某些活动监视审计迹的过程。可以编写一个脚本，以便检测到异常事件时，触发自动提升对特定用户或特定系统的审计。

例如，可以编写执行以下操作的脚本：

1. 在受审计系统上监视审计文件的创建。
2. 使用 `tail` 命令处理审计文件。

通过 `praudit` 命令对 `tail -0f` 命令进行管道输出，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程序。
或者，可以使用脚本来触发自动响应。
4. 经常监视审计文件系统，以查看是否有新的 `not_terminated` 审计文件出现。
5. 当等待中的 `tail` 进程的文件不再被写入信息时，终止这些进程。

管理审计（任务）

本章提供用于帮助您在 Oracle Solaris 系统上配置和管理审计的过程。本章还包含管理审计迹和对审计服务进行故障排除的相关说明。以下是本章中信息的列表。

- 第 501 页中的“管理审计（任务列表）”
- 第 502 页中的“配置审计服务（任务）”
- 第 515 页中的“配置审计日志（任务）”
- 第 523 页中的“在区域中配置审计服务（任务）”
- 第 526 页中的“启用和禁用审计服务（任务）”
- 第 530 页中的“在本地系统上管理审计记录（任务）”
- 第 539 页中的“审计服务的故障排除（任务）”

有关审计服务的概述，请参见第 26 章，[审计（概述）](#)。有关规划建议，请参见第 27 章，[规划审计](#)。有关参考信息，请参见第 29 章，[审计（参考）](#)。

管理审计（任务列表）

以下任务列表列出了管理审计所需执行的主要任务。除故障排除部分以外，对其他几项任务进行了排序。

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 491 页中的“规划审计（任务）”
2. 配置审计。	设置将为用户和系统记录哪些审计事件。（可选）修改审计策略、审计类-事件映射和队列控制。	第 502 页中的“配置审计服务（任务列表）”
	配置决定审计记录存储位置及格式的插件。	第 515 页中的“配置审计日志（任务）”

任务	说明	参考
3. 启用审计。	启动审计服务。	第 526 页中的“启用和禁用审计服务（任务）”
	停止审计服务。	
	在已安装非全局区域的主机上，每个区域运行一个审计服务。或者，这些区域使用全局区域审计服务。	第 523 页中的“在区域中配置审计服务（任务）”
4. 管理审计记录。	收集并分析审计迹中的审计数据。	第 530 页中的“在本地系统上管理审计记录（任务列表）”
对审计进行故障排除。	调试并解决审计服务问题。	第 539 页中的“审计服务的故障排除（任务）”

配置审计服务（任务）

在网络上启用审计之前，可以修改缺省值以满足站点审计要求。最佳做法是在第一批用户登录之前尽可能定制审计配置。

如果已实现区域，则可以选择从全局区域审计所有区域或单独审计非全局区域。有关概述的信息，请参见第 556 页中的“审计和 Oracle Solaris 区域”。有关规划的信息，请参见第 492 页中的“如何规划区域中的审计”。有关过程的信息，请参见第 523 页中的“在区域中配置审计服务（任务）”。

配置审计服务（任务列表）

以下任务列表列出了配置审计的过程。所有任务都是可选的。

任务	说明	参考
显示审计缺省值。	配置审计之前，显示缺省策略、队列控制、标志和插件用法。	第 503 页中的“如何显示审计服务缺省值”
选择要审计的事件。	预选系统范围的审计类。如果是可归属事件，则针对此事件审计所有用户。	第 504 页中的“如何预选审计类”
选择要针对特定用户审计的事件。	设置系统范围的审计类的用户特定例外。	第 505 页中的“如何配置用户审计特征”
指定审计策略。	定义站点所需的其他审计数据。	第 508 页中的“如何更改审计策略”
指定队列控制。	修改缺省缓冲区大小、队列中的审计记录以及两次将审计记录写入缓冲区之间的间隔。	第 511 页中的“如何更改审计队列控制”
创建 audit_warn 电子邮件别名。	定义需要关注审计服务时接收电子邮件警告的人员。	第 512 页中的“如何配置 audit_warn 电子邮件别名”

任务	说明	参考
配置审计日志。	为每个插件配置审计记录的位置。	第 515 页中的“配置审计日志（任务）”
添加审计类。	通过创建用来保存关键事件的新审计类来减少审计记录数目。	第 513 页中的“如何添加审计类”
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 514 页中的“如何更改审计事件的类成员身份”

▼ 如何显示审计服务缺省值

本过程中的命令显示当前审计配置。本过程中的输出是从未配置的系统获取的。

开始之前 您必须分配有 Audit Configuration（审计配置）或 Audit Control（审计控制）权限配置文件。

1 成为具有所需安全属性的管理员。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 显示可归属事件的预选类。

```
# auditconfig -getflags
active user default audit flags = lo(0x1000,0x1000)
configured user default audit flags = lo(0x1000,0x1000)
```

lo 是 login/logout 审计类的标志。掩码输出的格式为 (*success,failure*)。

3 显示无归属事件的预选类。

```
# auditconfig -getnaflags
active non-attributable audit flags = lo(0x1000,0x1000)
configured non-attributable audit flags = lo(0x1000,0x1000)
```

注 – 要查看哪些事件指定给了某一类以及正在记录哪些事件，请运行 `auditrecord -c class` 命令。

4 显示审计策略。

```
$ auditconfig -getpolicy
configured audit policies = cnt
active audit policies = cnt
```

活动策略是当前策略，但审计服务并不存储该策略值。已配置策略由审计服务存储，因此重新启动审计服务时会恢复该策略。

5 显示有关审计插件的信息。

```
$ auditconfig -getplugin
Plugin: audit_binfile (active)
  Attributes: p_dir=/var/audit;p_fsize=0;p_minfree=1;

Plugin: audit_syslog (inactive)
  Attributes: p_flags=;

Plugin: audit_remote (inactive)
  Attributes: p_hosts=;p_retries=3;p_timeout=5;
```

缺省情况下 audit_binfile 插件处于活动状态。

6 显示审计队列控制。

```
$ auditconfig -getqctrl
no configured audit queue hiwater mark
no configured audit queue lowwater mark
no configured audit queue buffer size
no configured audit queue delay
active audit queue hiwater mark (records) = 100
active audit queue lowwater mark (records) = 10
active audit queue buffer size (bytes) = 8192
active audit queue delay (ticks) = 20
```

活动队列控制是内核当前使用的队列控制。字符串 no configured 表示系统使用的是缺省值。

7 显示为现有用户预选的审计类。

查找用户，然后显示每个用户的 audit_flags 属性值。

```
# who
adoe pts/1 Oct 10 10:20 (:0.0)
adoe pts/2 Oct 10 10:20 (:0.0)
jdoe pts/5 Oct 12 12:20 (:0.0)
jdoe pts/6 Oct 12 12:20 (:0.0)
...
# userattr audit_flags adoe
# userattr audit_flags jdoe
```

缺省情况下，仅针对系统范围的设置审计用户。

有关 userattr 命令的说明，请参见 [userattr\(1\)](#) 手册页。有关 audit_flags 关键字的说明，请参见 [user_attr\(4\)](#) 手册页。

▼ 如何预选审计类

预选包含要监视事件的审计类。不记录预选类之外的事件。

开始之前 您必须分配有 Audit Configuration（审计配置）权限配置文件。

1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 确定当前预选类。

```
# auditconfig -getflags
...

# auditconfig -getnaflags
'''
```

有关输出的说明，请参见第 503 页中的“如何显示审计服务缺省值”。

3 预选可归属类。

```
# auditconfig -setflags lo,ps,fw
user default audit flags = ps,lo,fw(0x101002,0x101002)
```

此命令审计登录/注销、进程启动/停止以及文件写入类中的成功事件和失败事件。

注 - auditconfig -setflags 命令不向当前系统缺省值添加类。此命令替换系统缺省值，因此必须指定要预选的所有类。

4 预选无归属类。

na 类包含其他事件之间的 PROM、引导和无归属挂载。

```
# auditconfig -setnaflags lo,na
non-attributable audit flags = lo,na(0x1400,0x1400)
```

lo 和 na 是 -setnaflags 选项唯一有用的两个参数。

注 - auditconfig -setnaflags 命令替换系统缺省值，因此必须指定要预选的所有类。

▼ 如何配置用户审计特征

通过基于每个用户而非基于每个系统预选类，有时可以降低审计对系统性能的影响。此外，您可能还需要审计与系统稍有不同的特定用户。

为每个用户预选的审计类是由 audit_flags 安全属性指定的。这些用户特定值以及系统的预选类确定用户的审计掩码，如第 559 页中的“进程审计特征”中所述。

开始之前 您必须是 root 角色。

● 在 user_attr 或 prof_attr 数据库中设置审计标志。

■ 要为用户设置审计标志，请使用 usermod 命令。

```
# usermod -K audit_flags=fw:no jdoe
```

audit_flags 关键字的格式为 *always-audit:never-audit*。

always-audit 列出针对此用户审计的审计类。对系统范围类的修改带有插入记号 (^) 前缀。添加到系统范围类的类不带插入记号前缀。

never-audit 列出从不针对用户审计的审计类，即使在系统范围内审计这些审计事件也是如此。对系统范围类的修改带有插入记号 (^) 前缀。

要指定多个审计类，请使用逗号分隔类。有关更多信息，请参见 [audit_flags\(5\)](#) 手册页。

- 要为权限配置文件设置审计标志，请使用 **profiles** 命令。

```
# profiles -p "System Administrator"
profiles:System Administrator> set name="Audited System Administrator"
profiles:Audited System Administrator> set always_audit=fw,as
profiles:Audited System Administrator> end
profiles:Audited System Administrator> exit
```

将 Audited System Administrator（审计的系统管理员）权限配置文件指定给用户或角色时，按第 187 页中的“搜索指定安全属性的顺序”中所述的搜索顺序针对这些标志审计该用户或角色。

示例 28-1 更改对单个用户进行审计的事件

在本示例中，所有用户的审计预选掩码如下：

```
# auditconfig -getflags
active user default audit flags = ss,lo(0x11000,0x11000)
configured user default audit flags = ss,lo(0x11000,0x11000)
```

除管理员以外没有任何用户登录。

为了降低 AUE_PFEEXEC 审计事件对系统资源的影响，管理员将不在系统级别审计此事件。相反，管理员将为用户 jdoe 预选 pf 类。pf 类是在示例 28-10 中创建的。

```
# usermod -K audit_flags=pf:no jdoe
```

userattr 命令显示添加内容。

```
# userattr audit_flags jdoe
pf:no
```

用户 jdoe 登录时，jdoe 的审计预选掩码为 audit_flags 值和系统缺省值的组合。289 是 jdoe 登录 shell 的 PID。

```
# auditconfig -getpinfo 289
audit id = jdoe(1234)
process preselection mask = ss,pf,lo(0x0100000000000000,0x0100000008011000)
terminal id (maj,min,host) = 242,511,example1(192.168.160.171)
audit session id = 103203403
```

示例 28-2 修改单个用户的审计预选例外

在本示例中，所有用户的审计预选掩码如下：

```
# auditconfig -getflags
active user default audit flags = ss,lo(0x11000,0x11000)
configured user default audit flags = ss,lo(0x11000,0x11000)
```

除管理员以外没有任何用户登录。

管理员决定不为 jdoe 用户收集失败的 ss 事件。

```
# usermod -K audit_flags=~ss:no jdoe
```

userattr 命令显示例外。

```
# userattr audit_flags jdoe
^~ss:no
```

用户 jdoe 登录时，jdoe 的审计预选掩码为 audit_flags 值和系统缺省值的组合。289 是 jdoe 登录 shell 的 PID。

```
# auditconfig -getpinfo 289
audit id = jdoe(1234)
process preselection mask = +ss,lo(0x11000,0x1000)
terminal id (maj,min,host) = 242,511,example1(192.168.160.171)
audit session id = 103203403
```

示例 28-3 审计选定用户，非系统范围的审计

在本示例中，审计四个选定用户在系统上的登录和角色活动。没有为系统预选审计类。

首先，管理员删除所有系统范围的标志。

```
# auditconfig -setflags no
user default audit flags = no(0x0,0x0)
```

然后，管理员为这四个用户预选两个审计类。pf 类是在[示例 28-10](#)中创建的。

```
# usermod -K audit_flags=lo,pf:no jdoe
# usermod -K audit_flags=lo,pf:no kdoe
# usermod -K audit_flags=lo,pf:no pdoe
# usermod -K audit_flags=lo,pf:no zdoe
```

然后，管理员为 root 角色预选 pf 类。

```
# userattr audit_flags root
# rolemod -K audit_flags=lo,pf:no root
# userattr audit_flags root
lo,pf:no
```

为了继续记录未经授权的入侵，管理员将不更改无归属登录的审计。

```
# auditconfig -getnaflags
active non-attributable audit flags = lo(0x1000,0x1000)
configured non-attributable audit flags = lo(0x1000,0x1000)
```

示例 28-4 删除用户的审计标志

在以下示例中，管理员删除所有用户特定的审计标志。继续审计当前已登录用户的现有进程。

管理员运行 `usermod` 命令，但不为 `audit_flags` 关键字设置任何值。

```
# usermod -K audit_flags= jdoe
# usermod -K audit_flags= kdoe
# usermod -K audit_flags= ldoe
```

然后，管理员验证删除。

```
# userattr audit_flags jdoe
# userattr audit_flags kdoe
# userattr audit_flags ldoe
```

示例 28-5 为用户组创建权限配置文件

管理员需要站点的所有管理权限配置文件才能显式审计 `pf` 类。对于要指定的每个权限配置文件，管理员都会在包含审计标志的 LDAP 中创建特定于站点的版本。

首先，管理员克隆现有权限配置文件，然后更改名称并添加审计标志。

```
# profiles -p "Network Wifi Management" -S ldap
profiles: Network Wifi Management> set name="Wifi Management"
profiles: Wifi Management> set desc="Audited wifi management"
profiles: Wifi Management> set audit_always=pf
profiles: Wifi Management> exit
```

针对要使用的每个权限配置文件重复此过程之后，管理员将列出 `Wifi Management`（Wifi 管理）配置文件中的信息。

```
# profiles -p "Wifi Management" -S ldap info
name=Wifi Management
desc=Audited wifi management
auths=solaris.network.wifi.config
help=RtNetWifiMngmnt.html
always_audit=pf
```

▼ 如何更改审计策略

审计策略确定本地系统审计记录的特征。您可能会更改审计策略以记录审计命令的相关详细信息、为每个记录添加区域名称，或满足其他站点安全要求。

开始之前 您必须分配有 Audit Configuration（审计配置）权限配置文件。

- 1 成为具有所需安全属性的管理员。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。

- 2 查看当前审计策略。
`$ auditconfig -getpolicy`
...
有关输出的说明，请参见第 503 页中的“如何显示审计服务缺省值”。

- 3 查看可用的策略选项。
`$ auditconfig -lspolicy`

policy string	description:
ahlt	halt machine if it can not record an async event
all	all policies for the zone
arge	include exec environment args in audit recs
argv	include exec command line args in audit recs
cnt	when no more space, drop recs and keep a cnt
group	include supplementary groups in audit recs
none	no policies
path	allow multiple paths per event
perzone	use a separate queue and auditd per zone
public	audit public files
seq	include a sequence number in audit recs
trail	include trailer token in audit recs
windata_down	include downgraded window information in audit recs
windata_up	include upgraded window information in audit recs
zonename	include zonename token in audit recs

注 – 只能在全局区域中设置 `perzone` 和 `ahlt` 策略选项。有关使用特定策略选项的权衡，请参见第 496 页中的“了解审计策略”。

- 4 启用或禁用选定的审计策略选项。
`# auditconfig [ -t ] -setpolicy [prefix]policy[,policy...]`

-t	可选。创建临时或活动策略。可以设置用于调试或测试的临时策略。 刷新审计服务或通过 <code>auditconfig -setpolicy</code> 命令修改策略之前，临时策略起作用。
prefix	prefix 值 + 将策略列表添加到当前策略中。prefix 值 - 从当前策略中删除策略列表。没有前缀时，将重置审计策略。通过此选项，您可以保留当前审计策略。
policy	选择要启用或禁用的策略。

示例 28-6 设置 ahlt 审计策略选项

在本示例中，禁用了 cnt 策略，启用了 ahlt 策略。在此配置下，当审计队列已满而发生异步事件时，会停止系统的使用。发生同步事件时，会挂起创建线程的进程。安全性比可用性重要时，此配置适用。有关更多信息，请参见第 558 页中的“同步事件和异步事件的审计策略”。

```
# auditconfig -setpolicy -cnt
# auditconfig -setpolicy +ahlt
```

ahlt 策略前的加号 (+) 将策略添加到当前策略设置。没有加号时，ahlt 策略将替换当前策略设置。

示例 28-7 设置临时审计策略

在本示例中，启用了审计服务，且配置了 ahlt 审计策略。管理员将 trail 审计策略添加到活动策略中 (+trail)，但未将审计服务配置为永久使用 trail 审计策略 (-t)。trail 策略用于帮助恢复损坏的审计迹。

```
$ auditconfig -setpolicy ahlt
$ auditconfig -getpolicy
  configured audit policies = ahlt
  active audit policies = ahlt
$ auditconfig -t -setpolicy +trail
  configured audit policies = ahlt
  active audit policies = ahlt, trail
```

调试完成后，管理员将禁用 trail 策略。

```
$ auditconfig -setpolicy -trail
$ auditconfig -getpolicy
  configured audit policies = ahlt
  active audit policies = ahlt
```

通过运行 audit -s 命令刷新审计服务也可以删除此临时策略，以及审计服务中的任何其他临时值。有关其他临时值的示例，请参见第 511 页中的“如何更改审计队列控制”。

示例 28-8 设置 perzone 审计策略

在本示例中，将 perzone 审计策略添加到全局区域中的现有策略。由于 perzone 策略设置是作为永久属性存储的，因此 perzone 策略在会话过程中重新启动审计服务后才生效。

```
$ auditconfig -getpolicy
  configured audit policies = cnt
  active audit policies = cnt
$ auditconfig -setpolicy +perzone
$ auditconfig -getpolicy
```

```
configured audit policies = perzone,cnt
active audit policies = perzone,cnt
```

▼ 如何更改审计队列控制

审计服务提供审计队列参数的缺省值。可以使用 `auditconfig` 命令检查、更改和临时更改这些值。

开始之前 您必须分配有 Audit Configuration（审计配置）权限配置文件。

1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 查看审计队列控制的当前值。

```
$ auditconfig -getqctrl
...
```

有关输出的说明，请参见第 503 页中的“如何显示审计服务缺省值”。

3 修改选定的审计队列控制。

有关审计队列控制的示例和说明，请参见 `auditconfig(1M)` 手册页。

- 要修改部分或全部审计队列控制，请使用 `-setqctrl` 选项。

```
# auditconfig [ -t ] -setqctrl hiwater lowater bufsz interval
```

例如，将 *interval* 值设置为 10，但不设置其他控制。

```
# auditconfig -setqctrl 0 0 0 10
```

- 要修改特定审计队列控制，请指定其选项。`-setqdelay` 选项等同于 `-setqctrl 0 0 0 interval`，如 `# auditconfig -setqdelay 10` 中所示。

```
# auditconfig [ -t ] -setqhiwater value
# auditconfig [ -t ] -setqlowater value
# auditconfig [ -t ] -setqbufsz value
# auditconfig [ -t ] -setqdelay value
```

示例 28-9 将审计队列控制重置为缺省值

管理员设置所有审计队列控制，然后将系统信息库中的 *lowater* 值更改回缺省值。

```
# auditconfig -setqctrl 200 5 10216 10
# auditconfig -setqctrl 200 0 10216 10
configured audit queue hiwater mark (records) = 200
no configured audit queue lowater mark
configured audit queue buffer size (bytes) = 10216
configured audit queue delay (ticks) = 10
```

```
active audit queue hiwater mark (records) = 200
active audit queue lowater mark (records) = 5
active audit queue buffer size (bytes) = 10216
active audit queue delay (ticks) = 10
```

随后，管理员将当前会话的 *lowater* 值设置为缺省值。

```
# auditconfig -setqlowater 10
# auditconfig -getqlowater
configured audit queue lowater mark (records) = 10
active audit queue lowater mark (records) = 10
```

▼ 如何配置 **audit_warn** 电子邮件别名

`/etc/security/audit_warn` 脚本将生成邮件，通知管理员可能需要关注的审计事件。您可以定制脚本，可以将邮件发送给 `root` 以外的帐户。

如果设置了 `perzone` 策略，非全局区域的管理员必须在非全局区域中配置 `audit_warn` 电子邮件别名。

开始之前 您必须是 `root` 角色。

● 配置 **audit_warn** 电子邮件别名。

选择以下选项之一：

- **方法 1**—在 `audit_warn` 脚本中使用其他电子邮件帐户替换 `audit_warn` 电子邮件别名。

将脚本的 `ADDRESS` 行中的 `audit_warn` 电子邮件别名更改为其他地址：

```
#ADDRESS=audit_warn          # standard alias for audit alerts
ADDRESS=audadmin             # role alias for audit alerts
```



注意—升级到新版本的 Oracle Solaris OS 时，必须手动合并定制的 `audit_warn` 文件和 `audit_warn.new` 文件。此新文件可能会包含重要更改。有关升级时 `preserve=renamenew` 文件操作的说明，请参见 `pkg(5)` 手册页。

- **方法 2**—将 `audit_warn` 电子邮件重定向到其他电子邮件帐户。

在这种情况下，需要将 `audit_warn` 电子邮件别名添加到适当的电子邮件别名文件。可以将别名添加到本地 `/etc/mail/aliases` 文件或 `mail_aliases` 数据库的名称空间中。如果将 `root` 和 `audadmin` 电子邮件帐户添加为 `audit_warn` 电子邮件别名的成员，`/etc/mail/aliases` 条目将类似于以下内容：

```
audit_warn: root,audadmin
```

然后，运行 `newaliases` 命令为 `aliases` 文件重新生成随机访问数据库。

```
# newaliases
/etc/mail/aliases: 14 aliases, longest 10 bytes, 156 bytes total
```

▼ 如何添加审计类

创建您自己的审计类时，可以只将需要针对您所在站点审计的审计事件存放到该类中。

在一个系统上添加类时，将此更改复制到正在审计的所有系统中。最佳做法是先创建审计类，然后再启用审计服务。



注意 – 升级到新版本的 Oracle Solaris OS 时，必须手动合并定制的 `audit_class` 文件和 `audit_class.new` 文件。此新文件可能会包含重要更改。有关升级时 `preserve=renamew` 文件操作的说明，请参见 `pkg(5)` 手册页。

开始之前 条目必须是唯一的。您必须选择空闲位。用户可以使用的位在 `/etc/security/audit_class` 文件中有所介绍。

您必须是 `root` 角色。

- 1 可选保存 `audit_class` 文件的备份副本。

```
# cp /etc/security/audit_class /etc/security/audit_class.orig
```

- 2 在 `audit_class` 文件中添加新项。

每一项都具有以下格式：

```
0x64bitnumber:flag:description
```

有关字段的说明，请参见 [audit_class\(4\)](#) 手册页。有关现有类的列表，请阅读 `/etc/security/audit_class` 文件。

示例 28-10 创建新的审计类

本示例创建一个类来保存角色中执行的管理命令。`audit_class` 文件中添加的项如下所示：

```
0x0100000000000000:pf:profile command
```

该条目创建新的 `pf` 审计类。[示例 28-11](#) 将填充新审计类。

故障排除 如果定制了 `audit_class` 文件，请确保系统审计预选掩码的所有用户例外与新的审计类一致。`audit_flags` 值不是 `audit_class` 文件的子集时发生错误。

▼ 如何更改审计事件的类成员身份

可能需要更改审计事件的类成员身份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。



注意 – 切勿注释掉 `audit_event` 中的事件。该文件供 `praudit` 命令用来读取二进制审计文件。归档审计文件可能包含该文件中列出的事件。

在一个系统上重新配置审计事件到类的映射时，将此更改复制到正在审计的所有系统中。最佳做法是用户登录之前更改事件到类映射。



注意 – 升级到新版本的 Oracle Solaris OS 时，必须手动合并定制的 `audit_event` 文件和 `audit_event.new` 文件。此新文件可能会包含重要更改。有关升级时 `preserve=renamew` 文件操作的说明，请参见 `pkg(5)` 手册页。

开始之前 您必须是 `root` 角色。

- 1 可选保存 `audit_event` 文件的备份副本。

```
# cp /etc/security/audit_event /etc/security/audit_event.orig
```

- 2 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 通常是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 28-11 将现有审计事件映射到新类

本示例中将现有审计事件映射到在 [示例 28-10](#) 中创建的新类。缺省情况下，`AUE_PFEXEC` 审计事件映射到四个类：`ps`、`ex`、`ua` 和 `as`。通过创建新类，管理员可以审计 `AUE_PFEXEC` 事件而无需审计其他四类中的事件。

```
# grep pf /etc/security/audit_class
0x0100000000000000:pf:profile command
# vi /etc/security/audit_event
116:AUE_PFEXEC:execve(2) with pfexec enabled:pf
# auditconfig -setflags lo,pf
user default audit flags = pf,lo(0x0100000000001000,0x0100000000001000)
```

配置审计日志（任务）

两个审计插件（`audit_binfile` 和 `audit_syslog`）将审计日志发送到您可以配置的位置。以下任务帮助您配置这些日志。

配置审计日志（任务列表）

以下任务列表列出了为各种插件配置审计日志的过程。所有任务都是可选的。

任务	说明	参考
为 <code>audit_binfile</code> 插件添加本地存储。	为审计文件创建本地磁盘空间，并使用文件权限来保护它们。	第 515 页中的“如何为审计文件创建 ZFS 文件系统”
为 <code>audit_binfile</code> 插件指定存储。	标识用于二进制审计记录的目录。	第 518 页中的“如何为审计迹指定审计空间”
为 <code>audit_remote</code> 插件配置存储。	使您能够通过受保护的机制将审计记录发送到远程系统信息库。	第 520 页中的“如何向远程系统信息库发送审计文件”
为 <code>audit_syslog</code> 插件配置存储。	使您能够以文本格式将审计事件流式传输到 <code>syslog</code> 。	第 521 页中的“如何配置 <code>syslog</code> 审计日志”

▼ 如何为审计文件创建 ZFS 文件系统

以下过程说明如何为审计文件创建 ZFS 池以及相应的文件系统和挂载点。缺省情况下，`/var/audit` 文件系统为 `audit_binfile` 插件保存审计文件。

开始之前 您必须分配有 ZFS File System Management（ZFS 文件系统管理）和 ZFS Storage Management（ZFS 存储管理）权限配置文件。通过后一个配置文件可以创建存储池。

- 1 成为具有所需安全属性的管理员。
- 有关更多信息，请参见[第 150 页中的“如何获取管理权限”](#)。
- 2 确定所需的磁盘空间量。
- 每个主机至少指定 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。

注 - 缺省类预选在 `/var/audit` 中创建文件，对于 `lo` 类中记录的每个事件实例（如登录、注销或角色承担），这些文件增长约 80 字节。

3 创建镜像 ZFS 存储池。

`zpool create` 命令创建一个存储池作为 ZFS 文件系统的容器。有关更多信息，请参见《Oracle Solaris 管理：ZFS 文件系统》中的第 1 章“Oracle Solaris ZFS 文件系统（介绍）”。

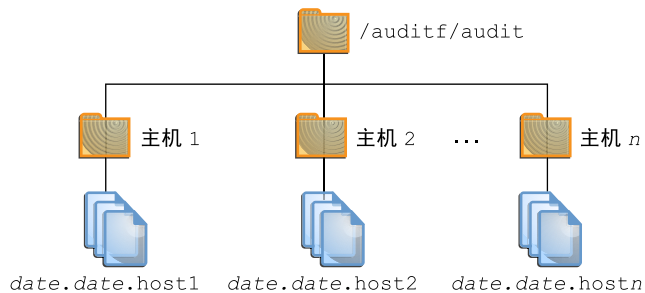
```
# zpool create audit-pool mirror disk1 disk2
```

例如，从两个磁盘（`c3t1d0` 和 `c3t2d0`）创建 `auditp` 池并镜像这两个磁盘。

```
# zpool create auditp mirror c3t1d0 c3t2d0
```

4 为审计文件创建 ZFS 文件系统和挂载点。

使用一个命令创建文件系统和挂载点。创建时，文件系统已挂载。例如，下图显示了按主机名存储的审计迹存储。



注- 如果计划加密文件系统，必须在创建文件系统时对其进行加密。有关示例，请参见示例 28-12。

加密需要进行管理。例如，挂载时需要口令短语。有关更多信息，请参见《Oracle Solaris 管理：ZFS 文件系统》中的“加密 ZFS 文件系统”。

```
# zfs create -o mountpoint=/mountpoint audit-pool/mountpoint
```

例如，为 `auditf` 文件系统创建 `/audit` 挂载点。

```
# zfs create -o mountpoint=/audit auditp/auditf
```

5 为审计文件创建 ZFS 文件系统。

```
# zfs create -p auditp/auditf/system
```

例如，为 `sys1` 系统创建未加密的 ZFS 文件系统。

```
# zfs create -p auditp/auditf/sys1
```


6 可选为审计文件创建其他文件系统。

创建其他文件系统的一个原因是阻止审计溢出。可以设置每个文件系统的 ZFS 配额，如 [步骤 9](#) 中所示。达到每个配额时，`audit_warn` 电子邮件别名会通知您。要释放空间，可以将关闭的审计文件移动到远程服务器。

```
# zfs create -p auditp/auditf/sys1.1
# zfs create -p auditp/auditf/sys1.2
```

7 保护父审计文件系统。

为池中的所有文件系统将以下 ZFS 属性设置为 `off`：

```
# zfs set devices=off auditp/auditf
# zfs set exec=off auditp/auditf
# zfs set setuid=off auditp/auditf
```

8 压缩池中的审计文件。

通常，压缩是在 ZFS 中的文件系统级别设置的。但是，由于此池中的所有文件系统都包含审计文件，因此会在顶级数据集为池设置压缩。

```
# zfs set compression=on auditp
```

另请参见《[Oracle Solaris 管理：ZFS 文件系统](#)》中的“ZFS 压缩、重复数据删除和加密属性之间的交互”。

9 设置配额。

可以设置父文件和/或后代文件系统上的配额。如果设置父审计文件系统上的配额，后代文件系统上的配额将强加其他限制。

a. 设置父审计文件系统上的配额。

在以下示例中，`auditp` 池中的两个磁盘都达到配额时，`audit_warn` 脚本将通知审计管理员。

```
# zfs set quota=510G auditp/auditf
```

b. 设置后代审计文件系统上的配额。

在以下示例中，达到 `auditp/auditf/system` 文件系统的配额时，`audit_warn` 脚本将通知审计管理员。

```
# zfs set quota=170G auditp/auditf/sys1
# zfs set quota=170G auditp/auditf/sys1.1
# zfs set quota=165G auditp/auditf/sys1.2
```

10 对于大型池，限制审计文件的大小。

缺省情况下，审计文件可以增长到池的大小。为了便于管理，限制审计文件的大小。请参见 [示例 28-14](#)。

示例 28-12 为审计文件创建加密文件系统

为了符合站点安全要求，管理员将在打开加密的情况下创建审计文件系统。然后，管理员设置挂载点。

```
# zfs create -o encryption=on auditp/auditf
Enter passphrase for auditp/auditf': /** Type 8-character minimum passphrase**/
Enter again: /** Confirm passphrase **/
# zfs set -o mountpoint=/audit auditp/auditf
```

管理员在 auditf 文件系统下创建其他文件系统时，也将加密这些后代文件系统。

示例 28-13 在 /var/audit 目录上设置配额

在本示例中，管理员在缺省审计文件系统上设置配额。达到此配额时，audit_warn 脚本将警告审计管理员。

```
# zfs set quota=252G rpool/var/audit
```

▼ 如何为审计迹指定审计空间

在本过程中，使用 audit_binfile 插件的属性将其他磁盘空间指定给审计迹。

开始之前 您必须分配有 Audit Configuration（审计配置）权限配置文件。

1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 确定 audit_binfile 插件的属性。

阅读 [audit_binfile\(5\)](#) 手册页的 OBJECT ATTRIBUTES 部分。

```
# man audit_binfile
...
OBJECT ATTRIBUTES
    The p_dir attribute specifies where the audit files will be
    created. The directories are listed in the order in which
    they are to be used.

    The p_minfree attribute defines the percentage of free space
    that the audit system requires before the audit daemon invokes
    the audit_warn script.

    The p_fsize attribute defines the maximum size in bytes that
    an audit file can become before it is automatically closed
    and a new audit file opened. ...
```

3 要将目录添加到审计迹，请指定 p_dir 属性。

缺省文件系统为 /var/audit。

```
# auditconfig -setplugin audit_binfile active p_dir=/audit/sys1.1,/var/audit
```

上述命令将 `/audit/sys1.1` 文件系统设置为审计文件的主目录，将缺省 `/var/audit` 文件系统设置为辅助目录。在此方案中，`/var/audit` 是最后考虑采用的目录。要成功完成此配置，`/audit/sys1.1` 文件系统必须存在。

在第 515 页中的“如何为审计文件创建 ZFS 文件系统”中已创建了类似的文件系统。

4 刷新审计服务。

`auditconfig -setplugin` 命令设置已配置的值。该值为审计服务的属性，因此刷新或重新启动该服务时会恢复该值。刷新或重新启动审计服务时，已配置的值变为活动状态。有关配置的活动值的信息，请参见 [auditconfig\(1M\)](#) 手册页。

```
# audit -s
```

示例 28-14 限制 audit_binfile 插件的文件大小

在以下示例中，二进制审计文件的大小设置为特定大小。大小是以兆字节为单位指定的。

```
# auditconfig -setplugin audit_binfile active p_fsize=4M
# auditconfig -getplugin audit_binfile
Plugin: audit_binfile (active)
Attributes: p_dir=/var/audit;p_fsize=4M;p_minfree=1;
```

缺省情况下，审计文件大小可以无限制地增长。为了创建较小的审计文件，管理员指定文件大小上限为 4 MB。达到大小限制时，审计服务将创建新文件。管理员刷新审计服务后，文件大小限制开始生效。

```
# audit -s
```

示例 28-15 指定对审计插件的多个更改

在以下示例中，具有高吞吐量和大型 ZFS 池的系统上的管理员将更改 `audit_binfile` 插件的队列大小、二进制文件大小和软限制警告。管理员允许审计文件增长到 4 GB，在剩余 2% 的 ZFS 池时收到警告，并将允许的队列大小翻倍。缺省队列大小是内核审计队列的高水位标志 100，如 `active audit queue hiwater mark (records) = 100` 中所示。

```
# auditconfig -getplugin audit_binfile
Plugin: audit_binfile (active)
Attributes: p_dir=/var/audit;p_fsize=2G;p_minfree=1;
# auditconfig -setplugin audit_binfile active "p_minfree=2;p_fsize=4G" 200
# auditconfig -getplugin audit_binfile
Plugin: audit_binfile (active)
Attributes: p_dir=/var/audit;p_fsize=4G;p_minfree=2;
Queue size: 200
```

管理员刷新审计服务后，更改的规范开始生效。

```
# audit -s
```

示例 28-16 删除审计插件的队列大小

在以下示例中，将删除 `audit_binfile` 插件的队列大小。

```
# auditconfig -getplugin audit_binfile
Plugin: audit_binfile (active)
  Attributes: p_dir=/var/audit;p_fsize=4G;p_minfree=2;
             Queue size: 200
# auditconfig -setplugin audit_binfile active "" ""
# auditconfig -getplugin audit_binfile
Plugin: audit_binfile (active)
  Attributes: p_dir=/var/audit;p_fsize=4G;p_minfree=2;
```

最后的空引号 ("") 将插件的队列大小设置为缺省值。

管理员刷新审计服务后，插件的 `qsize` 规范中的更改开始生效。

```
# audit -s
```

示例 28-17 设置警告的软限制

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空间只有 2% 时发出警告。

```
# auditconfig -setplugin audit_binfile active p_minfree=2
```

缺省百分比为 1 (1)。对于大型 ZFS 池，请适当地选择较低的百分比。例如，16 TB 池的 10% 大约是 16 GB，如果设为此值将在仍剩余大量磁盘空间时向审计管理员发出警告。如果值为 2，将在剩余大约 2 GB 磁盘空间时发送 `audit_warn` 消息。

`audit_warn` 电子邮件别名用于接收警告。要设置别名，请参见第 512 页中的“[如何配置 audit_warn 电子邮件别名](#)”。

对于大型池，管理员还会将文件大小限制为 3 GB。

```
# auditconfig -setplugin audit_binfile active p_fsize=3G
```

管理员刷新审计服务后，插件的 `p_minfree` 和 `p_fsize` 规范开始生效。

```
# audit -s
```

▼ 如何向远程系统信息库发送审计文件

在本过程中，使用 `audit_remote` 插件的属性将审计迹发送到远程审计系统信息库。

开始之前 远程系统信息库上必须具有审计服务的接收者。您必须分配有 Audit Configuration（审计配置）权限配置文件。

1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 确定 `audit_remote` 插件的属性。

阅读 `audit_remote(5)` 手册页的 OBJECT ATTRIBUTES 部分。

```
# man audit_remote
...
OBJECT ATTRIBUTES
    The p_hosts attribute specifies the remote servers.
    You can also specify the port number and the GSS-API
    mechanism.

    The p_retries attribute specifies the number of retries for
    connecting and sending data. The default is 3.

    The p_timeout attribute specifies the number of seconds
    in which a connection times out.
```

缺省端口是 `solaris_audit` IANA 指定的端口 16162/tcp。缺省机制是 `kerberos_v5`。缺省超时为 5 秒。您还可以指定插件的队列大小。

3 要指定远程主机，请使用 `p_hosts` 属性。

```
# auditconfig -setplugin audit_remote active p_hosts=rhost1:16088:kerberos_v5
```

4 要指定重试次数，请使用 `p_retries` 属性。

```
# auditconfig -setplugin audit_remote active p_retries=5
```

5 要指定连接超时长度，请使用 `p_timeout` 属性。

```
# auditconfig -setplugin audit_remote active p_timeout=3
```

6 刷新审计服务。

审计服务在刷新时将读取审计插件更改。

```
# audit -s
```

▼ 如何配置 `syslog` 审计日志

可以指示审计服务，将审计队列中的部分或全部审计记录复制到 `syslog` 实用程序。如果记录二进制审计数据和文本摘要，二进制数据将提供完整的审计记录，而摘要将筛选数据以供实时查看。

开始之前 要配置 `audit_syslog` 插件，必须分配有 Audit Configuration（审计配置）权限配置文件。要配置 `syslog` 实用程序，必须是 `root` 角色。

1 选择要发送给 `audit_syslog` 插件的审计类，并激活该插件。

注 -p_flags 审计类必须预选为系统缺省值，或在用户的审计标志或权限配置文件的审计标志中预选。不会为未预选的类收集记录。

```
# auditconfig -setplugin audit_syslog active p_flags=lo,+as,-ss
```

2 配置 syslog 实用程序。

a. 将 audit.notice 项添加到 syslog.conf 文件中。

此项包括日志文件的位置。

```
# cat /etc/syslog.conf
...
audit.notice      /var/adm/auditlog
```

b. 创建日志文件。

```
# touch /var/adm/auditlog
```

c. 刷新 syslog 服务的配置信息。

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

3 刷新审计服务。

审计服务在刷新时将读取审计插件的更改。

```
# audit -s
```

4 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 28-18 指定 syslog 输出的审计类

在以下示例中，syslog 实用程序收集预选的审计类的子集。pf 类是在 [示例 28-10](#) 中创建的。

```
# auditconfig -setnaflags lo,na
# auditconfig -setflags lo,ss
# usermod -K audit_flags=pf:no jdoe
# auditconfig -setplugin audit_syslog active p_flags=lo,+na,-ss,+pf
```

auditconfig 命令的参数指示系统收集所有登录/注销、无归属事件以及系统状态审计记录的更改。audit_syslog 插件项指示 syslog 实用程序收集所有登录、成功的无归属事件以及失败的系统状态更改。

对于 jdoe 用户，二进制审计记录包含对调用 pfexec 命令的所有使用。要使这些事件可供以后选择，audit_binfile 或 audit_remote 插件必须处于活动状态。syslog 实用程序收集对 pfexec 命令的成功调用。

示例 28-19 将 syslog 审计记录放置在远程系统上

可以更改 `syslog.conf` 文件中的 `audit.notice` 项，使其指向远程系统。在本示例中，本地系统的名称为 `sys1.1`。远程系统是 `remote1`。

```
sys1.1 # cat /etc/syslog.conf
...
audit.notice      @remote1
```

在 `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
```

在区域中配置审计服务（任务）

审计服务审计整个系统，包括区域中的审计事件。安装了非全局区域的系统可以以相同方式审计所有区域，也可基于每个区域配置审计。有关背景信息，请参见第 487 页中的“在具有 Oracle Solaris 区域的系统上审计”。有关规划信息，请参见第 492 页中的“如何规划区域中的审计”。

如果按照与审计全局区域完全相同的方法审计非全局区域，审计服务将在全局区域中运行。该服务从全局区域和所有非全局区域收集审计记录。非全局区域管理员可能无权访问审计记录。

注 – 全局区域管理员可以选择修改非全局区域中的用户的审计掩码。

单独审计非全局区域时，单个审计服务在审计的每个区域中运行。每个区域都收集自己的审计记录。记录对于非全局区域和非全局区域根中的全局区域均可见。

▼ 如何配置以相同方式审计所有区域

此过程可以实现以相同方式审计所有区域。该方法需要的计算机开销和管理资源最少。

开始之前 您必须是 `root` 角色。

1 配置全局区域的审计。

完成第 502 页中的“配置审计服务（任务列表）”中的任务，但注意以下例外：

- 不要启用 `perzone` 审计策略。
- 不要启用审计服务。请在完成非全局区域的审计配置后启用审计服务。

- 设置 `zonename` 策略。此策略将区域名称添加到每个审计记录。

```
# auditconfig -setpolicy +zonename
```

2 如果修改了审计配置文件，请将这些文件从全局区域复制到每个非全局区域。

如果修改了 `audit_class` 或 `audit_event` 文件，请采用以下两种方式之一复制它：

- 可以回送挂载文件。
- 可以复制文件。

非全局区域必须在运行中。

- 将更改的 `audit_class` 和 `audit_event` 文件作为回送文件系统 (lofs) 进行挂载。

a. 从全局区域中，中止非全局区域。

```
# zoneadm -z non-global-zone halt
```

b. 为在全局区域中修改的每个审计配置文件创建一个只读回送挂载。

```
# zonecfg -z non-global-zone
add fs
  set special=/etc/security/audit-file
  set dir=/etc/security/audit-file
  set type=lofs
  add options [ro,nodevices,nosetuid]
  commit
end
exit
```

c. 要使更改生效，请引导该非全局区域。

```
# zoneadm -z non-global-zone boot
```

之后，如果在全局区域中修改了审计配置文件，请重新引导区域以刷新非全局区域中回送挂载的文件。

- 复制文件。

a. 从全局区域中，列出非全局区域中的 `/etc/security` 目录。

```
# ls /zone/zonename/root/etc/security/
```

b. 将更改的 `audit_class` 和 `audit_event` 文件复制到区域的 `/etc/security` 目录。

```
# cp /etc/security/audit-file /zone/zonename/root/etc/security/audit-file
```

之后，如果在全局区域中更改其中一个文件，必须将文件重新复制到非全局区域。

在全局区域中启用审计服务时，将审计非全局区域。

示例 28-20 在区域中挂载作为回送挂载的审计配置文件

在本示例中，系统管理员修改了 `audit_class`、`audit_event` 和 `audit_warn` 文件。

`audit_warn` 文件仅在全局区域中读取，因此不必将其挂载到非全局区域。

在系统 `machine1` 中，管理员创建了两个非全局区域：`machine1-webserver` 和 `machine1-appserver`。管理员已经完成了对审计配置文件的修改。如果管理员以后修改文件，必须重新引导区域以重新读取回送挂载。

```
# zoneadm -z machine1-webserver halt
# zoneadm -z machine1-appserver halt
# zonecfg -z machine1-webserver
add fs
  set special=/etc/security/audit_class
  set dir=/etc/security/audit_class
  set type=lofs
  add options [ro,nodevices,noexec,uid]
  commit
end
add fs
  set special=/etc/security/audit_event
  set dir=/etc/security/audit_event
  set type=lofs
  add options [ro,nodevices,noexec,uid]
  commit
end
exit
# zonecfg -z machine1-appserver
add fs
  set special=/etc/security/audit_class
  set dir=/etc/security/audit_class
  set type=lofs
  add options [ro,nodevices,noexec,uid]
  commit
end
...
exit
```

重新引导非全局区域时，`audit_class` 和 `audit_event` 文件在区域中为只读。

▼ 如何配置每区域审计

此过程使每个区域的管理员可以在他们的各自区域中控制审计服务。有关策略选项的完整列表，请参见 [auditconfig\(1M\)](#) 手册页。

开始之前 您必须分配有 Audit Configuration（审计配置）权限配置文件才能配置审计。您必须分配有 Audit Control（审计控制）权限配置文件才能启用审计服务。

1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

- 2 在全局区域中，配置审计。
 - a. 完成第 502 页中的“配置审计服务（任务列表）”中的任务。
 - b. 添加 **perzone** 审计策略。有关命令，请参见示例 28-8。

注-您无需在全局区域中启用审计服务。

- 3 在计划审计的每个非全局区域中，配置审计文件。
 - a. 完成第 502 页中的“配置审计服务（任务列表）”中的任务。
 - b. 不要配置系统范围的审计设置。
具体来说，不要将 **perzone** 或 **ahlt** 策略添加到非全局区域中。

- 4 在您的区域中启用审计。

```
myzone# audit -s
```

示例 28-21 在非全局区域中禁用审计

本示例适用于全局区域设置了 **perzone** 审计策略的情况。**noaudit** 区域的管理员禁用该区域的审计。

```
noauditzone # auditconfig -getcond
audit condition = auditing
noauditzone # audit -t
noauditzone # auditconfig -getcond
audit condition = noaudit
```

启用和禁用审计服务（任务）

审计服务在缺省情况下处于启用状态，且由 **auditconfig** 命令配置。如果在全局区域中设置 **perzone** 审计策略，区域管理员可以在其非全局区域中启用、刷新和禁用服务。

▼ 如何刷新审计服务

启用审计服务后，如果更改了审计插件的配置，此过程将更新审计服务。

开始之前 您必须分配有 Audit Control（审计控制）权限配置文件。

- 1 成为具有所需安全属性的管理员。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 刷新审计服务。

```
# audit -s
```

注 – 刷新审计服务时，将丢失所有临时配置设置。审计策略和队列控制允许临时设置。有关更多信息，请参见 [auditconfig\(1M\)](#) 手册页。

3 更新当前正在审计的用户的预选掩码。

系统基于与每个进程相关联的审计预选掩码生成审计记录。刷新审计服务不会更改现有进程的掩码。要显式重置现有进程的预选掩码，请参见第 547 页中的“[如何更新已登录用户的预选掩码](#)”。

示例 28–22 刷新启用的审计服务

在本示例中，管理员重新配置审计、验证更改，然后刷新审计服务。

- 首先，管理员添加临时策略。

```
# auditconfig -t -setpolicy +zonename
# auditconfig -getpolicy
configured audit policies = ahlt,arge,argv,perzone
active audit policies = ahlt,arge,argv,perzone,zonename
```

- 然后，管理员指定队列控制。

```
# auditconfig -setqctrl 200 20 0 0
# auditconfig -getqctrl
configured audit queue hiwater mark (records) = 200
configured audit queue lowater mark (records) = 20
configured audit queue buffer size (bytes) = 8192
configured audit queue delay (ticks) = 20
active audit queue hiwater mark (records) = 200
active audit queue lowater mark (records) = 20
active audit queue buffer size (bytes) = 8192
active audit queue delay (ticks) = 20
```

- 随后，管理员指定插件属性。

- 对于 audit_binfile 插件，管理员将删除 qsize 值。

```
# auditconfig -getplugin audit_binfile
Plugin: audit_binfile (active)
  Attributes: p_dir=/audit/sys1.1,/var/audit;
              p_minfree=2;p_fsize=4G;
Queue size: 200
# auditconfig -setplugin audit_binfile active "" ""
# auditconfig -getplugin audit_binfile
Plugin: audit_binfile (active)
  Attributes: p_dir=/audit/sys1.1,/var/audit
              p_minfree=2;p_fsize=4G;
```

最后的空引号("") 将插件的队列大小设置为缺省值。

- 对于 audit_syslog 插件，管理员将指定要发送到 syslog 的成功登录和注销事件以及失败的可执行文件。此插件的 qsize 设置为 50。

```
# auditconfig -setplugin audit_syslog active p_flags=+lo,-ex 50
# auditconfig -getplugin audit_syslog
auditconfig -getplugin audit_syslog
Plugin: audit_syslog (active)
  Attributes: p_flags=+lo,-ex;
  Queue size: 50
```

- 管理员不配置也不使用 `audit_remote` 插件。
- 然后，管理员刷新审计服务并验证配置。
- 不再设置临时 `zonename` 策略。

```
# audit -s
# auditconfig -getpolicy
configured audit policies = ahlt,arge,argv,perzone
active audit policies = ahlt,arge,argv,perzone
```

- 队列控制保留原样。

```
# auditconfig -getqctrl
configured audit queue hiwater mark (records) = 200
configured audit queue lowater mark (records) = 20
configured audit queue buffer size (bytes) = 8192
configured audit queue delay (ticks) = 20
active audit queue hiwater mark (records) = 200
active audit queue lowater mark (records) = 20
active audit queue buffer size (bytes) = 8192
active audit queue delay (ticks) = 20
```

- `audit_binfile` 插件没有指定的队列大小。`audit_syslog` 插件具有指定的队列大小。

```
# auditconfig -getplugin
Plugin: audit_binfile (active)
  Attributes: p_dir=/var/audit;p_fsize=4G;p_minfree=2;

Plugin: audit_syslog (active)
  Attributes: p_flags=+lo,-ex;
  Queue size: 50
...
```

▼ 如何禁用审计服务

本过程显示设置了 `perzone` 审计策略后如何在全局区域和非全局区域中禁用审计。

- 如果没有设置 `perzone` 审计策略，将禁用所有区域的审计。
- 如果在全局区域中设置了 `perzone` 审计策略，该策略在启用了审计的非全局区域中保持有效。

由于在全局区域中设置了 `perzone` 策略，因此非全局区域将继续在全局区域重新引导和非全局区域重新引导时收集审计记录。

开始之前 您必须分配有 Audit Control（审计控制）权限配置文件。

- 1 成为具有所需安全属性的管理员。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。
- 2 运行 **audit -t** 命令以禁用审计服务。
有关更多信息，请参见 [audit\(1M\)](#) 和 [auditd\(1M\)](#) 手册页。
 - 在全局区域中，禁用审计服务。

```
# audit -t
```

 如果没有设置 `perzone` 审计策略，此命令将禁用所有区域中的审计。
 - 在非全局区域中，禁用审计服务。
 如果设置了 `perzone` 审计策略，非全局区域管理员必须在非全局区域中禁用服务。

```
zone1 # audit -t
```

▼ 如何启用审计服务

本过程在管理员禁用审计服务后为所有区域启用该服务。要在非全局区域中启动审计服务，请参见[示例 28-23](#)。

开始之前 要启用或禁用审计服务，您必须分配有 Audit Control（审计控制）权限配置文件。

- 1 成为具有所需安全属性的管理员。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。
- 2 使用 **audit -s** 命令启用审计服务。

```
# audit -s
```

 有关更多信息，请参见 [audit\(1M\)](#) 手册页。
- 3 验证是否已启用审计。

```
# auditconfig -getcond
audit condition = auditing
```

示例 28-23 在非全局区域中启用审计

在本示例中，区域管理员执行以下操作后，为 `zone1` 启用审计服务：

- 全局区域管理员在全局区域中设置 `perzone` 策略。
- 非全局区域的区域管理员配置审计服务和每用户定制。

然后，区域管理员为区域启用审计服务。

```
zone1# audit -s
```

在本地系统上管理审计记录（任务）

缺省插件 `audit_binfile` 创建审计迹。通过管理审计迹，可以监视网络中的用户操作。审计可以生成大量数据。以下任务说明如何处理所有这些数据。

在本地系统上管理审计记录（任务列表）

以下任务列表列出了有关选择、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信息的显示顺序。	第 530 页中的“如何显示审计记录定义”
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迹中。	第 532 页中的“如何合并审计迹中的审计文件”
选择要检查的记录。	选择要研究的特定事件。	第 533 页中的“如何从审计迹中选择审计事件”
显示审计记录。	可以查看二进制审计记录。	第 535 页中的“如何查看二进制审计文件的内容”
清除错误指定的审计文件。	向审计服务意外打开的审计文件提供结束时间戳。	第 537 页中的“如何清除 <code>not_terminated</code> 审计文件”
防止审计迹溢出。	防止写满审计文件系统。	第 538 页中的“如何防止审计迹溢出”

▼ 如何显示审计记录定义

`auditrecord` 命令显示审计记录定义。定义提供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的定义放置在 HTML 文件中。
-a 选项用于列出所有审计事件定义。-h 选项使列表采用 HTML 格式。

```
% auditrecord -ah > audit.events.html
```

提示 – 在浏览器中显示 HTML 文件时，请使用浏览器的“查找”工具来查找特定审计记录定义。

有关更多信息，请参见 [auditrecord\(1M\)](#) 手册页。

示例 28-24 显示程序的审计记录格式

在本示例中，将显示 login 程序生成的所有审计记录的格式。登录程序包括 rlogin、telnet、newgrp 和 Oracle Solaris 的安全 Shell 功能。

```
% auditrecord -p login
...
login: logout
  program      various          See login(1)
  event ID     6153             AUE_logout
  class        lo               (0x0000000000001000)
...
newgrp
  program      newgrp           See newgrp login
  event ID     6212             AUE_newgrp_login
  class        lo               (0x0000000000001000)
...
rlogin
  program      /usr/sbin/login   See login(1) - rlogin
  event ID     6155             AUE_rlogin
  class        lo               (0x0000000000001000)
...
/usr/lib/ssh/sshd
  program      /usr/lib/ssh/sshd See login - ssh
  event ID     6172             AUE_ssh
  class        lo               (0x0000000000001000)
...
telnet login
  program      /usr/sbin/login   See login(1) - telnet
  event ID     6154             AUE_telnet
  class        lo               (0x0000000000001000)
...
```

示例 28-25 显示审计类的审计记录格式

在本示例中，显示在示例 28-10 中创建的 pf 类中所有审计记录的格式。

```
% auditrecord -c pf

pfexec
  system call  pfexec           See execve(2) with pfexec enabled
  event ID    116              AUE_PFEXEC
  class       pf               (0x0100000000000000)
    header
    path       pathname of the executable
    path       pathname of working directory
    [privileges] privileges if the limit or inheritable set are changed
    [privileges] privileges if the limit or inheritable set are changed
    [process]   process if ruid, euid, rgid or egid is changed
    exec_arguments
    [exec_environment] output if arge policy is set
    subject
    [use_of_privilege]
    return
```

使用特权时，将记录 `use_of_privilege` 标记。更改限制或可继承权限集时，将记录 `privileges` 标记。更改 ID 时，将记录 `process` 标记。无需任何策略选项即可将这些标记包含在记录中。

▼ 如何合并审计迹中的审计文件

通过合并所有审计目录中的所有审计文件，可以分析整个审计迹的内容。`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。如果没有指定路径，`auditreduce` 命令将使用 `/var/audit` 文件系统。

注 – 由于审计迹中的时间戳采用的是协调通用时间 (Coordinated Universal Time, UTC)，因此日期和小时必须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。

开始之前 您必须分配有 Audit Review（审计查看）权限配置文件。

1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 创建用于存储合并的审计文件的文件系统。

该文件系统应该位于与您在第 515 页中的“如何为审计文件创建 ZFS 文件系统”中创建的文件系统不同的 `zpool` 中，以便存储原始文件。

3 合并审计迹中的审计记录。

转到用于存储合并的审计文件的目录。从此目录中，将审计记录合并到具有指定后缀的文件中。将合并本地系统上审计迹中的所有目录。

```
# cd audit-storage-directory
# auditreduce -Uppercase-option -O suffix
```

`auditreduce` 命令的大写选项用于处理审计迹中的文件。大写选项包括：

- A 选择审计迹中的所有文件。
- C 只选择完整文件。
- M 选择带特定后缀的文件。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开始时间和结束时间为 14 个字符时间戳且后缀为 *suffix* 的审计文件。
- R *pathname* 指定该选项可读取备用审计根目录 *pathname* 中的审计文件。
- S *server* 指定该选项可读取指定服务器中的审计文件。

有关选项的完整列表，请参见 `auditreduce(1M)` 手册页。

4 将合并的文件移动到其他 zpool 中的文件系统。

要将文件移动到其他系统，请使用 `sftp` 命令。有关说明，请参见 [sftp\(1\)](#) 手册页。

示例 28-26 将审计文件复制到摘要文件

在以下示例中，分配有 System Administrator（系统管理员）权限配置文件的管理人员将审计迹中的所有文件复制到其他文件系统上的合并文件中。`/var/audit/storage` 文件系统位于 `/var/audit` 文件系统（审计根文件系统）的独立磁盘上。

```
$ cd /var/audit/storage
$ auditreduce -A -O All
$ ls /var/audit/storage/*All
20100827183214.20100827215318.All
```

在以下示例中，仅将完整文件从审计迹复制到合并文件中。完整路径指定为 `-O` 选项的值。路径的最后组成部分 `Complete` 用作后缀。

```
$ auditreduce -C -O /var/audit/storage/Complete
$ ls /var/audit/storage/*Complete
20100827183214.20100827214217.Complete
```

在以下示例中，仅将完整文件从 `sys1.1` 系统复制到合并文件中。

```
$ cd /var/audit/storage
$ auditreduce -M sys1.1 -O example1summ
$ ls /var/audit/storage/*summ
20100827183214.20100827214217.example1summ
```

示例 28-27 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其他位置时将其删除。在以下示例中，将 `sys1.1` 系统的完整审计文件复制到 `audit_summary` 文件系统，便于以后检查。

```
$ cd /var/audit/audit_summary
$ auditreduce -C -O daily_sys1.1 -D sys1.1
$ ls *sys1.1
20100827183214.20100827214217.daily_sys1.1
```

此命令成功完成后，将删除 `sys1.1` 系统中的审计文件，这些审计文件是 `*daily_sys1.1` 文件的输入。

▼ 如何从审计迹中选择审计事件

可以过滤审计记录以进行检查。有关过滤选项的完整列表，请参见 [auditreduce\(1M\)](#) 手册页。

开始之前 您必须分配有 Audit Review（审计查看）权限配置文件。

- 1 成为具有所需安全属性的管理员。
有关更多信息，请参见第 150 页中的“如何获取管理权限”。

- 2 从审计迹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` 选择归属于特定用户的所有事件。*argument* 是用户名。另一个用户选项 `-e` 选择归属于有效用户 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。

有关选项的完整列表，请参见 [auditreduce\(1M\)](#) 手册页。

示例 28-28 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太感兴趣的记录。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注销记录。如果需要获取完整的审计迹，可以从备份介质中恢复。

```
# cd /var/audit/audit_summary
# auditreduce -O lo.summary -b 20100827 -c lo; compress *lo.summary
```

示例 28-29 将一个用户的审计记录复制到摘要文件

在本示例中，合并审计迹中包含特定用户名称的记录。`-e` 选项用于查找有效用户。`-u` 选项查找登录用户。

```
$ cd /var/audit/audit_summary
$ 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 2010 年 9 月 7 日（您的当地时间）用户登录和注销的时间。只检查那些以用户名作为文件后缀的文件。日期的简洁形式为 `yyyymmdd`。

```
# auditreduce -M tamiko -O tamikolo -d 20100907 -u tamiko -c lo
```

示例 28-30 将选定记录复制到单个文件中

在本示例中，将从审计迹中选择特定日期的登录和注销记录。将这些记录合并到一个目标文件中。将目标文件写入到包含审计根目录的文件系统以外的文件系统。

```
# auditreduce -c lo -d 20100827 -O /var/audit/audit_summary/logins
# ls /var/audit/audit_summary/*logins
/var/audit/audit_summary/20100827183936.20100827232326.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。

开始之前 您必须分配有 Audit Review（审计查看）权限配置文件。

- 1 成为具有所需安全属性的管理员。**

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

- 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` 命令以短格式显示审计记录，每行一个标记。使用 `-l` 选项在一行放置一条记录。

```
$ auditreduce -c lo | praudit -s
header,69,2,AUE_screenlock,,mach1,2010-10-14 08:02:56.348 -07:00
subject,jdoe,root,staff,jdoe,staff,856,50036632,82 0 mach1
return,success,0
sequence,1298
```

- `praudit -r` 命令以原始格式显示审计记录，每行一个标记。使用 `-l` 选项在一行放置一条记录。

```
$ auditreduce -c lo | praudit -r
21,69,2,6222,0x0000,10.132.136.45,1287070091,698391050
36,26700,0,10,26700,10,856,50036632,82 0 10.132.136.45
39,0,0
47,1298
```

- `praudit -x` 命令以 XML 格式显示审计记录，每行一个标记。使用 `-l` 选项在 XML 输出中的每一行放置一条记录。以下列表分为两行输出以适合此打印页面：

```
$ auditreduce -c lo | praudit -x
<record version="2" event="screenlock - unlock" host="mach1"
  iso8601="2010-10-14 08:28:11.698 -07:00">
<subject audit-uid="jdoe" uid="root" gid="staff" ruid="jdoe
  rgid="staff" pid="856" sid="50036632" tid="82 0 mach1"/>
<return errval="success" retval="0"/>
```

```
<sequence seq-num="1298"/>
</record>
```

示例 28-31 打印整个审计迹

通过打印命令的管道，整个审计迹的输出将传输到打印机。出于安全考虑，打印机的访问权限将受到限制。

```
# auditreduce | praudit | lp -d example.protected.printer
```

示例 28-32 查看特定审计文件

在本示例中，将在终端窗口中查看登录文件摘要。

```
# cd /var/audit/audit_summary/logins
# praudit 20100827183936.20100827232326.logins | more
```

示例 28-33 以 XML 格式显示审计记录

在本示例中，将审计记录转换为 XML 格式。

```
# praudit -x 20100827183214.20100827215318.logins > 20100827.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，以提取相关信息。

示例 28-34 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" \
    Finds the user-specified string
| tr '\002' '\012' \
    Restores the original newline breaks
```

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。

故障排除 如果出现类似如下的消息，表明您没有足够的特权，无法使用 `praudit` 命令：

```
praudit: Can't assign 20090408164827.20090408171614.sys1.1 to stdin.
```

在配置文件 `shell` 中运行 `praudit` 命令。您必须分配有 Audit Review（审计查看）权限配置文件。

▼ 如何清除 `not_terminated` 审计文件

发生异常系统中断时，如果审计服务的审计文件仍处于打开状态，则审计服务将存在。或者，文件系统无法访问，强制系统切换到新文件系统。在这种情况下，尽管审计文件不再用于审计记录，但此文件仍然保留，并以字符串 `not_terminated` 作为结束时间戳。使用 `auditreduce -O` 命令可为此文件提供正确时间戳。

开始之前 您必须分配有 Audit Review（审计查看）权限配置文件。

1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 在审计文件系统中，按创建顺序列出带有 `not_terminated` 字符串的文件。

```
# ls -Rlt audit-directory*/ * | 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。

-l 将文件列成一列。

3 清除旧的 `not_terminated` 文件。

对 `auditreduce -O` 命令指定旧文件的名称。

```
# auditreduce -O system-name old-not-terminated-file
```

4 删除旧的 `not_terminated` 文件。

```
# rm system-name old-not-terminated-file
```

示例 28-35 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，然后删除原始文件。

```
ls -Rlt */ * | grep not_terminated
.../egret.1/20100908162220.not_terminated.egret
.../egret.1/20100827215359.not_terminated.egret
# cd */egret.1
# auditreduce -O egret 20100908162220.not_terminated.egret
# ls -lt
```

```

20100908162220.not_terminated.egret      Current audit file
20100827230920.20100830000909.egret      Cleaned up audit file
20100827215359.not_terminated.egret      Input (old) audit file
# rm 20100827215359.not_terminated.egret
# ls -lt
20100908162220.not_terminated.egret      Current audit file
20100827230920.20100830000909.egret      Cleaned up audit file

```

新文件的开始时间戳反映 `not_terminated` 文件中第一个审计事件的时间。结束时间戳反映此文件中最后一个审计事件的时间。

▼ 如何防止审计迹溢出

如果安全策略要求保存所有审计数据，请防止审计记录丢失。

开始之前 您必须是 `root` 角色。

1 设置 `audit_binfile` 插件上的最小空闲大小。

使用 `p_minfree` 属性。

磁盘空间满足最小空闲大小时，`audit_warn` 电子邮件别名将发送警告。请参见[示例 28-17](#)。

2 设置计划安排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。此外，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。

3 设置计划安排以从审计文件系统中删除已归档的审计文件。

4 保存和存储辅助信息。

归档解释审计记录与审计迹所需的信息。至少应保存 `passwd`、`group` 和 `hosts` 文件。也可以归档 `audit_event` 和 `audit_class` 文件。

5 保留已归档审计文件的记录。

6 正确存储归档介质。

7 减少启用 ZFS 压缩所需的文件系统容量。

在专用于审计文件的 ZFS 文件系统上，压缩将大大缩小文件。有关示例，请参见[第 549 页](#)中的“[如何压缩专用文件系统上的审计文件](#)”。

另请参见《Oracle Solaris 管理：ZFS 文件系统》中的“ZFS 压缩、重复数据删除和加密属性之间的交互”。

- 8 通过创建摘要文件，减少存储的审计数据量。
- 可以使用 `auditreduce` 命令的选项从审计迹中提取摘要文件。摘要文件只包含指定类型审计事件的记录。要提取摘要文件，请参见示例 28-28 和示例 28-30。

审计服务的故障排除（任务）

本节介绍各种审计错误消息、首选项以及其他工具提供的审计。这些过程可以帮助您记录所需的审计事件和调试审计问题。

审计服务的故障排除（任务列表）

以下任务列表列出了审计的故障排除过程。

问题	解决方法	参考
为何配置了审计，却没有记录审计记录？	对审计服务进行故障排除。	第 540 页中的“如何确定审计正在运行”
怎样才能减少收集的审计信息量？	只审计您想要审计的事件。	第 542 页中的“如何减少生成的审计记录数目”
如何对用户在上系统上执行的所有操作进行审计？	针对每个命令审计一个或多个用户。	第 544 页中的“如何审计用户执行的所有命令”
如何更改正在记录的审计事件，并使更改对现有会话产生影响？	更新用户的预选掩码。	第 547 页中的“如何更新已登录用户的预选掩码”
如何找到对特定文件的修改？	审计文件修改，然后使用 <code>auditreduce</code> 命令找出特定文件。	第 545 页中的“如何找到对特定文件更改的审计记录”
如何减小审计文件的大小？	限制二进制审计文件的大小。	第 549 页中的“如何限制二进制审计文件的大小”
如何使审计文件使用更少的文件系统空间？	使用 ZFS 配额和压缩。	第 549 页中的“如何压缩专用文件系统上的审计文件”
如何从 <code>audit_event</code> 中删除审计事件？	正确更新 <code>audit_event</code> 文件。	第 548 页中的“如何阻止审计特定事件”
如何对 Oracle Solaris 系统的所有登录进行审计？	对从任何系统的登录进行审计。	第 550 页中的“如何审计从其他操作系统的登录”

问题	解决方法	参考
为什么没有为我的 FTP 传输保留审计记录？	对生成自己日志的实用程序使用适当的审计工具。	第 551 页中的“如何审计 FTP 和 SFTP 文件传输”

▼ 如何确定审计正在运行

缺省情况下启用审计。如果您认为审计尚未禁用，但没有将任何审计记录发送到活动插件，请使用以下过程隔离问题。

开始之前 要修改系统文件，您必须是 root 角色。要配置审计，必须为您指定 Audit Configuration（审计配置）权限配置文件。

1 确定审计正在运行中。

使用以下任一方法：

■ 检验当前审计条件。

以下列表说明，审计未在运行中：

```
# auditconfig -getcond
audit condition = noaudit
```

以下列表说明审计正在运行中：

```
# auditconfig -getcond
audit condition = auditing
```

■ 验证审计服务是否正在运行。

以下列表说明，审计未在运行中：

```
# svcs -x auditd
svc:/system/auditd:default (Solaris audit daemon)
State: disabled since Sun Oct 10 10:10:10 2010
Reason: Disabled by an administrator.
See: http://sun.com/msg/SMF-8000-05
See: auditd(1M)
See: audit(1M)
See: auditconfig(1M)
See: audit_flags(5)
See: audit_binfile(5)
See: audit_syslog(5)
See: audit_remote(5)
See: /var/svc/log/system-auditd:default.log
Impact: This service is not running.
```

以下列表说明，审计服务正在运行中：

```
# svcs auditd
STATE      STIME      FMRI
online     10:10:10  svc:/system/auditd:default
```


如果审计服务没有运行，请启用该服务。有关过程，请参见第 529 页中的“如何启用审计服务”。

2 验证是否至少有一个插件处于活动状态。

```
# audit -v
```

如果没有任何插件处于活动状态，则激活一个插件。

```
# auditconfig -setplugin audit_binfile active
```

3 如果创建了定制审计类，请确认已经将事件分配给该类。

例如，以下标志列表包含 Oracle Solaris 软件不提供的 pf 类：

```
# auditconfig -getflags
active user default audit flags = pf,lo(0x0100000000000000,00x0100000000001000)
configured user default audit flags = pf,lo(0x0100000000000000,00x0100000000001000)
```

有关创建 pf 类的说明，请参见第 513 页中的“如何添加审计类”。

a. 检验是否在 audit_class 文件中定义了该类。

审计类必须定义，其掩码必须唯一。

```
# grep pf /etc/security/audit_class      Verify class exists
0x0100000000000000:pf:profile
# grep 0x08000000 /etc/security/audit_class  Ensure mask is unique
0x0100000000000000:pf:profile
```

替换不唯一的掩码。如果没有定义该类，请定义它。否则，使用有效值运行 auditconfig -setflags 命令以重置当前标志。

b. 检验是否已经将事件分配给该类。

使用以下方法之一：

```
# auditconfig -lseven | egrep " pf|,pf|pf,"
AUE_PFEEXEC      116 pf execve(2) with pfexec enabled
```

```
# auditrecord -c pf
List of audit events assigned to pf class
```

如果未将事件分配给该类，请向其分配适当的事件。

4 如果前面的步骤仍未找出问题所在，请检查电子邮件和日志文件。

a. 读取发送到 audit_warn 别名的电子邮件。

audit_warn 脚本将警报消息发送到 audit_warn 电子邮件别名。如果缺少正确配置的别名，消息将发送到 root 帐户。

b. 检查审计服务的日志文件。

svcs -s auditd 命令的输出列出审计服务生成的审计日志的完整路径。有关示例，请参见步骤 1 中的列表。

c. 检查系统日志文件。

`audit_warn` 脚本将 `daemon.alert` 消息写入 `/var/log/syslog` 文件。
`/var/adm/messages` 文件可能会包含信息。

5 找到并解决问题后，启用或重新启动审计服务。

```
# audit -s
```

▼ 如何减少生成的审计记录数目

确定了要在您的站点上对哪些事件进行审计后，请参考以下建议创建可管理的审计文件。

开始之前 要预选审计类和设置审计策略，您必须分配有 **Audit Configuration**（审计配置）权限配置文件。要修改系统文件并将审计标志指定给用户、角色和权限配置文件，您必须是 `root` 角色。

1 使用缺省审计策略。

具体而言，应避免向审计迹中添加事件和审计标记。以下策略会增长审计迹的大小。

- `arge` 策略—向 `execv` 审计事件添加环境变量。
- `argv` 策略—向 `execv` 审计事件添加命令参数。
- `public` 策略—如果正在审计文件事件，每次 **public object**（公共对象）发生可审计事件时，都会将一个事件添加到审计迹中。文件类包括 `fa`、`fc`、`fd`、`fm`、`fr`、`fw` 和 `cl`。有关公共文件的定义，请参见第 478 页中的“审计术语和概念”。
- `path` 策略—将 `path` 标记添加到包含可选 `path` 标志的审计事件。
- `group` 策略—将组标记添加到包含可选 `newgroups` 标记的审计事件。
- `seq` 策略—将序列标记添加到每个审计事件。
- `trail` 策略—将尾部标记添加到每个审计事件。
- `windata_down` 策略—在配置了 `Trusted Extensions` 的系统上，在有标签窗口中的信息降级时添加事件。
- `windata_up` 策略—`在配置了 Trusted Extensions 的系统上，在有标签窗口中的信息升级后添加事件。`
- `zonename` 策略—将区域名称添加到每个审计事件。如果全局区域是配置的唯一区域，将字符串 `zone`、`global` 添加到每个审计事件。

以下审计记录显示了 `ls` 命令的用法。正在审计 `ex` 类，且正在使用缺省策略：

```
header,129,2,AUE_EXECVE,,mach1,2010-10-14 11:39:22.480 -07:00
path,/usr/bin/ls
attribute,100555,root,bin,21,320271,18446744073709551615
```

```
subject,jdoe,root,root,root,root,2404,50036632,82 0 mach1
return,success,0
```

以下是启用所有策略时的相同记录：

```
header,1578,2,AUE_EXECVE,,mach1,2010-10-14 11:45:46.658 -07:00
path,/usr/bin/ls
attribute,100555,root,bin,21,320271,18446744073709551615
exec_args,2,ls,/etc/security
exec_env,49,MANPATH=/usr/share/man,USER=jdoe,GDM_KEYBOARD_LAYOUT=us,EDITOR=gedit,
LANG=en_US.UTF-8,GDM_LANG=en_US.UTF-8,PS1=#,GDMSESSION=gnome,SESSIONTYPE=1,SHLVL=2,
HOME=/home/jdoe,LOGNAME=jdoe,G_FILENAME_ENCODING=@locale,UTF-8, PRINTER=example-dbl,
...
path,/lib/ld.so.1
attribute,100755,root,bin,21,393073,18446744073709551615
subject,jdoe,root,root,root,root,2424,50036632,82 0 mach1
group,root,other,bin,sys,adm,uucp,mail,tty,lp,nuucp,daemon
return,success,0
zone,global
sequence,197
trailer,1578
```

2 使用 `audit_syslog` 插件将部分审计事件发送到 `syslog`。

但不将这些审计事件发送到 `audit_binfile` 或 `audit_remote` 插件。只有当不要求保留发送到 `syslog` 日志的审计事件的二进制记录时，才适合使用此方法。

3 设置更少的系统范围审计标志并审计各个用户。

通过减少系统范围内审计的审计类的数目，来减少对所有用户的审计量。

使用 `roleadd`、`rolemod`、`useradd` 和 `usermod` 命令的 `audit_flags` 关键字审计特定用户和角色的事件。有关示例，请参见 [示例 28-18](#) 和 [usermod\(1M\)](#) 手册页。

使用 `profiles` 命令的 `always_audit` 和 `never_audit` 属性审计特定权限配置文件的事件。有关信息，请参见 [profiles\(1\)](#) 手册页。

注 - 与其他安全属性一样，审计标志会受到搜索顺序的影响。有关更多信息，请参见 [第 187 页](#) 中的“搜索指定安全属性的顺序”。

4 创建定制审计类。

可以在您的站点上创建审计类。仅将需要监视的审计事件分配给这些类。有关过程，请参见 [第 513 页](#) 中的“如何添加审计类”。



注意 - 如果修改了现有审计类分配，在升级到 Oracle Solaris OS 的新版本时，可能会保留所做的修改。但是，Oracle Solaris 中的新版本文件可能会包含必须手动并入安装的更改。请认真查看安装日志。有关更多信息，请参见 `pkg(5)` 手册页中 `preserve=renamenew` 的说明。

▼ 如何审计用户执行的所有命令

作为站点安全策略的一部分，有些站点会需要 root 帐户和管理角色运行的所有命令的审计记录。一些站点可能会需要所有用户运行的所有命令的审计记录。此外，站点可能还会要求记录命令参数和环境。

开始之前 要预选审计类和设置审计策略，您必须分配有 Audit Configuration（审计配置）权限配置文件。要将审计标志指定给用户、角色和权限配置文件，您必须是 root 角色。

1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 审计 lo 和 ex 类。

ex 类审计对 exec() 和 execve() 函数的所有调用。

lo 类审计登录、注销和屏幕锁定。以下输出列出了 ex 和 lo 类中的所有事件。

```
% auditconfig -lsevent | grep " lo "
AUE_login          6152 lo login - local
AUE_logout         6153 lo logout
AUE_telnet         6154 lo login - telnet
AUE_rlogin         6155 lo login - rlogin
AUE_rshd           6158 lo rsh access
AUE_su             6159 lo su
AUE_rexecd         6162 lo rexecd
AUE_passwd         6163 lo passwd
AUE_rexd           6164 lo rexd
AUE_ftp            6165 lo ftp access
AUE_ftp_logout     6171 lo ftp logout
AUE_ssh            6172 lo login - ssh
AUE_role_login     6173 lo role login
AUE_newgrp_login   6212 lo newgrp login
AUE_admin_authenticate 6213 lo admin login
AUE_screenlock     6221 lo screenlock - lock
AUE_screenunlock   6222 lo screenlock - unlock
AUE_zlogin         6227 lo login - zlogin
AUE_su_logout      6228 lo su logout
AUE_role_logout    6229 lo role logout
AUE_smbd_session   6244 lo smbd(lm) session setup
AUE_smbd_logoff    6245 lo smbd(lm) session logoff
AUE_ClientConnect  9101 lo client connection to x server
AUE_ClientDisconnect 9102 lo client disconn. from x server
% auditconfig -lsevent | egrep " ex |,ex |ex,"
AUE_EXECVE         23 ex,ps execve(2)
```

■ 要审计管理角色的这些类，请修改角色的安全属性。

在以下示例中，root 是一个角色。站点已创建 sysadm、auditadm 和 netadm 三个角色。针对 ex 和 lo 类中的成功事件和失败事件审计所有角色。

```
# rolemod -K audit_flags=lo,ex:no root
# rolemod -K audit_flags=lo,ex:no sysadm
# rolemod -K audit_flags=lo,ex:no auditadm
# rolemod -K audit_flags=lo,ex:no netadm
```

- 要审计所有用户的这些类，请设置系统范围的标志。

```
# auditconfig -setflags lo,ex
```

输出内容类似如下：

```
header,129,2,AUE_EXECVE,,mach1,2010-10-14 12:17:12.616 -07:00
path,/usr/bin/ls
attribute,100555,root,bin,21,320271,18446744073709551615
subject,jdoe,root,root,root,root,2486,50036632,82 0 mach1
return,success,0
```

- 3 要记录命令参数，请添加 **argv** 策略。

```
# auditconfig -setpolicy +argv
```

exec_args 标记用于记录命令参数：

```
header,151,2,AUE_EXECVE,,mach1,2010-10-14 12:26:17.373 -07:00
path,/usr/bin/ls
attribute,100555,root,bin,21,320271,18446744073709551615
exec_args,2,ls,/etc/security
subject,jdoe,root,root,root,root,2494,50036632,82 0 mach1
return,success,0
```

- 4 要记录命令的运行环境，请添加 **arge** 策略。

```
# auditconfig -setpolicy +arge
```

exec_env 标记用于记录命令环境：

```
header,1460,2,AUE_EXECVE,,mach1,2010-10-14 12:29:39.679 -07:00
path,/usr/bin/ls
attribute,100555,root,bin,21,320271,18446744073709551615
exec_args,2,ls,/etc/security
exec_env,49,MANPATH=/usr/share/man,USER=jdoe,GDM_KEYBOARD_LAYOUT=us,EDITOR=gedit,
LANG=en_US.UTF-8,GDM_LANG=en_US.UTF-8,PS1=#,GDMSESSION=gnome,SESSIONTYPE=1,SHLV=2,
HOME=/home/jdoe,LOGNAME=jdoe,G_FILENAME_ENCODING=@locale,UTF-8,
PRINTER=example-dbl,...,=/usr/bin/ls
subject,jdoe,root,root,root,root,2502,50036632,82 0 mach1
return,success,0
```

▼ 如何找到对特定文件更改的审计记录

如果您的目标是记录对有限数目的文件（如 `/etc/passwd` 和 `/etc/default` 目录下的文件）的写入情况，请使用 **auditreduce** 命令找到这些文件。

开始之前 您必须分配有 Audit Configuration（审计配置）权限配置文件才能使用 **auditconfig** 命令。您必须分配有 Audit Review（审计查看）权限配置文件才能使用 **auditreduce** 命令。要将审计标志指定给用户和角色，您必须是 **root** 角色。

- 1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 审计 fw 类。

将类添加到用户或角色的审计标志时生成的记录少于将类添加到系统范围的审计预选掩码时生成的记录。执行下列步骤之一：

- 将 fw 类添加到特定角色。

```
# rolemod -K audit_flags=fw:no root
# rolemod -K audit_flags=fw:no sysadm
# rolemod -K audit_flags=fw:no auditadm
# rolemod -K audit_flags=fw:no netadm
```

- 将 fw 类添加到系统范围的标志。

```
# auditconfig -getflags
active user default audit flags = lo(0x1000,0x1000)
configured user default audit flags = lo(0x1000,0x1000)
# auditconfig -setflags lo,fw
user default audit flags = lo,fw(0x1002,0x1002)
```

3 或者，审计成功的文件写入。

审计成功事件时生成的记录少于同时审计失败事件和成功事件时生成的记录。执行下列步骤之一：

- 将 +fw 标志添加到特定角色。

```
# rolemod -K audit_flags=+fw:no root
# rolemod -K audit_flags=+fw:no sysadm
# rolemod -K audit_flags=+fw:no auditadm
# rolemod -K audit_flags=+fw:no netadm
```

- 将 +fw 标志添加到系统范围的标志。

```
# auditconfig -getflags
active user default audit flags = lo(0x1000,0x1000)
configured user default audit flags = lo(0x1000,0x1000)
# auditconfig -setflags lo,+fw
user default audit flags = lo,+fw(0x1002,0x1000)
```

- 如果为成功事件和失败事件审计系统范围的标志，请为特定用户和角色设置例外。

```
# auditconfig -getflags
active user default audit flags = lo,fw(0x1002,0x1002)
configured user default audit flags = lo,fw(0x1002,0x1002)
# rolemod -K audit_flags=~fw:no root
# rolemod -K audit_flags=~fw:no sysadm
# rolemod -K audit_flags=~fw:no auditadm
# rolemod -K audit_flags=~fw:no netadm
```

系统范围的标志仍保持不变，但这四个角色的预选掩码发生了更改。

```
# auditconfig -getflags
active user default audit flags = lo,fw(0x1002,0x1000)
configured user default audit flags = lo,fw(0x1002,0x1000)
```

- 4 要找到特定文件的审计记录，请使用 **auditreduce** 命令。

```
# auditreduce -o file=/etc/passwd,/etc/default -O filechg
```

auditreduce 命令在审计迹中搜索所有 **file** 参数实例。该命令创建一个后缀为 **filechg** 的二进制文件，此文件包含了包括相关文件路径名的所有记录。有关 **-o file=pathname** 选项的语法，请参见 **auditreduce(1M)** 手册页。

- 5 要读取 **filechg** 文件，请使用 **praudit** 命令。

```
# praudit *filechg
```

▼ 如何更新已登录用户的预选掩码

您希望审计已登录用户系统范围的审计预选掩码的更改。

开始之前 您必须分配有 Audit Configuration（审计配置）权限配置文件。要停止用户会话，您必须分配有 Process Management（进程管理）权限配置文件。

- 1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

- 2 更新已登录用户的预选掩码。

您有两种选择。可以停止现有会话，或使用 **auditconfig** 命令更新预选掩码。

- 终止用户的现有会话。

用户可以注销后重新登录。或者，角色分配有 Process Management（进程管理）权限配置文件时，您可以手动停止（中止）活动会话。新的会话将继承新的预选掩码。但是，终止用户可能不现实。

- 动态更改每个已登录用户的预选掩码。

角色包含 Audit Configuration（审计配置）权限配置文件时，假设您已将系统范围的审计预选掩码从 **lo** 更改为 **lo,ex**。

```
# auditconfig -setflags lo,ex
```

- a. 列出登录的一般用户及其进程 ID。

```
# who -a
jdoe - vt/2          Jan 25 07:56 4:10 1597 (:0)
jdoe + pts/1         Jan 25 10:10 . 1706 (:0.0)
...
jdoe + pts/2         Jan 25 11:36 3:41 1706 (:0.0)
```

- b. 为便于以后进行比较，显示每个用户的预选掩码。

```
# auditconfig -getpinfo 1706
audit id = jdoe(1234)
```

```
process preselection mask = lo(0x1000,0x1000)
terminal id (maj,min,host) = 9426,65559,mach1(192.168.123.234)
audit session id = 103203403
```

c. 修改用户的预选掩码。

```
# auditconfig -setumask jdoe lo,ex      /* for this user */

# auditconfig -setsmask 103203403 lo,ex  /* for this session */

# auditconfig -setpmask 1706 lo,ex      /* for this process */
```

d. 验证用户的预选掩码是否已更改。

例如，检查更改掩码之前存在的进程。

```
# auditconfig -getpinfo 1706
audit id = jdoe(1234)
process preselection mask = ex,lo(0x40001000,0x40001000)
terminal id (maj,min,host) = 9426,65559,mach1(192.168.123.234)
audit session id = 103203403
```

▼ 如何阻止审计特定事件

出于维护目的，有时某个站点需要阻止对事件进行审计。

开始之前 您必须是 root 角色。

1 将相关事件的类更改为 no 类。

例如，事件 26 和 27 属于 pm 类。

```
## audit_event file
...
25:AUE_VFORK:vfork(2):ps
26:AUE_SETGROUPS:setgroups(2):pm
27:AUE_SETPGRP:setpgrp(2):pm
28:AUE_SWAPON:swapon(2):no
...
```

将这些事件更改为 no 类。

```
## audit_event file
...
25:AUE_VFORK:vfork(2):ps
26:AUE_SETGROUPS:setgroups(2):no
27:AUE_SETPGRP:setpgrp(2):no
28:AUE_SWAPON:swapon(2):no
...
```

如果当前正在审计 pm 类，现有会话仍将审计事件 26 和 27。要停止审计这些事件，必须遵循第 547 页中的“[如何更新已登录用户的预选掩码](#)”中的说明更新用户的预选掩码。



注意 – 切勿注释掉 `audit_event` 中的事件。该文件供 `praudit` 命令用来读取二进制审计文件。归档审计文件可能包含该文件中列出的事件。

2 刷新内核事件。

```
# auditconfig -conf
Configured 283 kernel events.
```

▼ 如何限制二进制审计文件的大小

系统对二进制审计文件的增大没有限制。但为了归档和搜索方便，您可能需要限制其大小。还可以基于原始文件创建较小的二进制文件。

开始之前 您必须分配有 Audit Configuration（审计配置）权限配置文件才能设置 `p_fsize` 属性。您必须分配有 Audit Review（审计查看）权限配置文件才能使用 `auditreduce` 命令。

1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 使用 `p_fsize` 属性限制各个二进制审计文件的大小。

有关 `p_fsize` 属性的说明，请参见 `audit_binfile(5)` 手册页的 OBJECT ATTRIBUTES 部分。

有关示例，请参见示例 28-14。

3 使用 `auditreduce` 命令选择记录，并将这些记录写入更小的文件供进一步分析。

`auditreduce -lowercase` 选项用于查找特定记录。

`auditreduce -Uppercase` 选项用于将您的选择写入文件。有关更多信息，请参见 `auditreduce(1M)` 手册页。

▼ 如何压缩专用文件系统上的审计文件

审计文件可以增大。您可以设置文件大小的上限，如示例 28-14 中所示。在本过程中，使用压缩来减小大小。

开始之前 您必须分配有 ZFS File System Management（ZFS 文件系统管理）和 ZFS Storage Management（ZFS 存储管理）权限配置文件。通过后一个配置文件可以创建存储池。

1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 将 ZFS 文件系统专用于审计文件。

有关过程，请参见第 515 页中的“如何为审计文件创建 ZFS 文件系统”。

3 使用以下选项之一压缩 ZFS 存储池。

使用这两个选项均可压缩审计文件系统。刷新审计服务后，将显示压缩率。

要设置压缩，请使用 `zfs set compression=on dataset` 命令。在以下示例中，ZFS 池 `auditp/auditf` 为数据集。

■ 使用缺省压缩算法。

```
# zfs set compression=on auditp/auditf
# audit -s
# zfs get compressratio auditp/auditf
NAME                PROPERTY          VALUE    SOURCE
auditp/auditf       compressratio     4.54x    -
```

■ 使用更高压缩算法。

```
# zfs set compression=gzip-9 auditp/auditf
# zfs get compression auditp/auditf
NAME                PROPERTY          VALUE    SOURCE
auditp/auditf       compression       gzip-9    local
# audit -s
# zfs get compressratio auditp/auditf
NAME                PROPERTY          VALUE    SOURCE
auditp/auditf       compressratio     16.89x    -
```

gzip-9 压缩算法导致文件占用的空间比缺省压缩算法 `lzjb` 少三分之一。有关更多信息，请参见《Oracle Solaris 管理：ZFS 文件系统》中的第 6 章“管理 Oracle Solaris ZFS 文件系统”。

▼ 如何审计从其他操作系统的登录

Oracle Solaris OS 可以审计所有登录，无论登录源为何。

开始之前 您必须分配有 Audit Configuration（审计配置）权限配置文件。

1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 审计可归属事件和无归属事件的 `lo` 类。

该类审计登录、注销和屏幕锁定。缺省情况下审计这些类。

```
# auditconfig -getflags
active user default audit flags = lo(0x1000,0x1000)
configured user default audit flags = lo(0x1000,0x1000)
# auditconfig -getnaflags
active non-attributable audit flags = lo(0x1000,0x1000)
configured non-attributable audit flags = lo(0x1000,0x1000)
```

3 如果值已更改，添加 **lo** 标志。

```
# auditconfig -getflags
active user default audit flags = as,st(0x20800,0x20800)
configured user default audit flags = as,st(0x20800,0x20800)
# auditconfig -setflags lo,as,st
user default audit flags = as,lo,st(0x21800,0x21800)
# auditconfig -getnaflags
active non-attributable audit flags = na(0x400,0x400)
configured non-attributable audit flags = na(0x400,0x400)
# auditconfig -setnaflags lo,na
non-attributable audit flags = lo,na(0x1400,0x1400)
```

注-要审计 ssh 登录，您的系统必须正在 Oracle Solaris 中运行 ssh 守护进程。在 Oracle Solaris 系统上，将针对该审计服务修改此守护进程。有关更多信息，请参见第 271 页中的“安全 Shell 和 OpenSSH 项目”。

▼ 如何审计 FTP 和 SFTP 文件传输

FTP 服务创建其文件传输的日志。通过预选 **ft** 审计类可以审计 ssh 协议下运行的 SFTP 服务。可以审计到这两种服务的登录。

开始之前 您必须分配有 Audit Configuration（审计配置）权限配置文件。

1 成为具有所需安全属性的管理员。

有关更多信息，请参见第 150 页中的“如何获取管理权限”。

2 要记录 FTP 服务的命令和文件传输，请参见 **proftpd(8)** 手册页。

有关可用的记录选项，请阅读“记录功能”部分。特别是，**log commands** 和 **log transfers** 选项可以提供有用的日志。

3 要记录 **sftp** 访问和文件传输，请审计 **ft** 类。

ft 类包含以下 SFTP 事务：

```
% auditrecord -c ft
file transfer: chmod ...
file transfer: chown ...
file transfer: download ...
file transfer: mkdir ...
file transfer: upload ...
file transfer: remove ...
file transfer: rename ...
file transfer: rmdir ...
file transfer: session start ...
file transfer: session end ...
file transfer: symlink ...
file transfer: utimes
```

4 要记录对FTP服务器的访问，请审计lo类。

如以下输出所示，登录和注销ftpd守护进程都会生成审计记录。

```
% auditrecord -c lo | more
...
in.ftpd
  program    /usr/sbin/in.ftpd    See ftp access
  event ID   6165                AUE_ftpd
  class      lo                  (0x0000000000001000)
    subject
    [text]                      error message
    return

in.ftpd
  program    /usr/sbin/in.ftpd    See ftp logout
  event ID   6171                AUE_ftpd_logout
  class      lo                  (0x0000000000001000)
    subject
    return
...
```

审计（参考）

本章介绍重要的审计组件。以下是本章中参考信息的列表。

- 第 553 页中的“审计服务”
- 第 554 页中的“审计服务手册页”
- 第 555 页中的“用于管理审计的权限配置文件”
- 第 556 页中的“审计和 Oracle Solaris 区域”
- 第 556 页中的“审计类”
- 第 557 页中的“审计插件”
- 第 558 页中的“审计策略”
- 第 559 页中的“进程审计特征”
- 第 560 页中的“审计迹”
- 第 560 页中的“二进制审计文件名称约定”
- 第 560 页中的“审计记录结构”
- 第 561 页中的“审计标记格式”

有关审计的概述，请参见第 26 章，[审计（概述）](#)。有关规划建议，请参见第 27 章，[规划审计](#)。有关在站点上配置审计的过程，请参见第 28 章，[管理审计（任务）](#)。

审计服务

缺省情况下，启用审计服务 `auditd`。要启用、刷新或禁用该服务，请参见第 526 页中的[“启用和禁用审计服务（任务）”](#)。

如果没有客户配置，则使用以下缺省配置：

- 审计所有登录事件。
审计成功和不成功的登录尝试。
- 审计所有用户的登录和注销事件，包括角色承担和屏幕锁定。
- `audit_binfile` 插件处于活动状态。`/var/audit` 目录存储审计记录，不限制审计文件的大小，且队列大小为 100 条记录。

- 设置了 `cnt` 策略。
审计记录填满可用磁盘空间时，系统将跟踪丢弃的审计记录数。剩余百分之一的可用磁盘空间时，将会发出警告。
- 设置了以下审计队列控制：
 - 生成记录锁定之前审计队列中的最大记录数－100
 - 阻塞的审计进程解除阻塞之前审计队列中的最小记录数－10
 - 审计队列的缓冲区大小－8192 字节
 - 将审计记录写入审计迹的时间间隔－20 秒

要显示缺省值，请参见第 503 页中的“如何显示审计服务缺省值”。

使用审计服务，可以设置临时值或活动值。这些值可能不同于已配置的值或属性值。

- 刷新或重新启动审计服务时，不会恢复临时值。
审计策略和审计队列控件均接受临时值。审计标志没有临时值。
- 已配置的值存储为服务的属性值，因此刷新或重新启动审计服务时会恢复这些值。

权限配置文件控制可以管理审计服务的人员。有关更多信息，请参见第 555 页中的“用于管理审计的权限配置文件”。

缺省情况下，以相同方式审计所有区域。请参见第 556 页中的“审计和 Oracle Solaris 区域”。

审计服务手册页

下表汇总了审计服务的主要管理手册页。

手册页	汇总
audit(1M)	控制审计服务的操作的命令 <code>audit -n</code> 为 <code>audit_binfile</code> 插件启动新的审计文件。 <code>audit -s</code> 启用并刷新审计。 <code>audit -t</code> 禁用审计。 <code>audit -v</code> 验证是否至少一个插件处于活动状态。
audit_binfile(5)	缺省审计插件，可将审计记录发送到二进制文件。另请参见第 557 页中的“审计插件”。
audit_remote(5)	将审计记录发送到远程接收者的审计插件。
audit_syslog(5)	将审计记录的文本摘要发送到 <code>syslog</code> 实用程序的审计插件。
audit_class(4)	包含审计类定义的文件。八个高序位可供客户创建新的审计类使用。有关在系统升级时修改此文件的影响，请参见第 513 页中的“如何添加审计类”。

手册页	汇总
audit_event(4)	包含审计事件定义且将事件映射到审计类的文件。可对此映射进行修改。有关在系统升级时修改此文件的影响，请参见第 514 页中的“如何更改审计事件的类成员身份”。
audit_flags(5)	介绍审计类预选的语法、仅选择失败事件或成功事件的前缀，以及修改现有预选的前缀。
audit.log(4)	介绍二进制审计文件的命名、文件的内部结构以及每个审计标记的结构。
audit_warn(1M)	审计服务在写入审计记录过程中遇到异常情况时，向某个电子邮件别名发出通知的脚本。您可以针对自己的站点定制此脚本，以便在可能需要手动干预时发出警告。也可以指定如何自动处理这些情况。
auditconfig(1M)	检索和设置审计配置参数的命令。 键入不带选项的 <code>auditconfig</code> ，以获取可检索和设置的参数列表。
auditrecord(1M)	显示 <code>/etc/security/audit_event</code> 文件中的审计事件定义的命令。有关示例输出，请参见第 530 页中的“如何显示审计记录定义”。
auditreduce(1M)	后选及合并以二进制格式存储的审计记录的命令。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，这些记录保持二进制格式。 大写选项会影响文件选择。小写选项会影响记录选择。
auditstat(1M)	显示内核审计统计信息的命令。例如，此命令可以显示内核审计队列中的记录数、丢弃的记录数以及由于系统调用而由用户进程在内核中生成的审计记录数。
praudit(1M)	从标准输入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的命令。可从 <code>auditreduce</code> 命令或者单个审计文件或审计文件列表中传输输入。对于当前审计文件，还可以使用 <code>tail -0f</code> 命令生成输入。 有关示例输出，请参见第 535 页中的“如何查看二进制审计文件的内容”。
syslog.conf(4)	配置为将审计记录的文本摘要要发送到 <code>audit_syslog</code> 插件的 <code>syslog</code> 实用程序的文件。

用于管理审计的权限配置文件

Oracle Solaris 提供了用于配置审计服务、启用和禁用服务以及分析审计迹的权限配置文件。要编辑审计配置文件，需要具有 `root` 特权。

- **"Audit Configuration" (审计配置)** —使管理员可以配置审计服务的参数并运行 `auditconfig` 命令。
- **"Audit Control" (审计控制)** —使管理员可以启动、刷新和禁用审计服务，并运行 `audit` 命令以启动、刷新或停止该服务。

- **"Audit Review" (审计查看)** — 使管理员可以分析审计记录。此权限配置文件可授权使用 `praudit` 和 `auditreduce` 命令读取审计记录。该管理员还可以运行 `auditstat` 命令。
- **"System Administrator" (系统管理员)** — 包括 "Audit Review" (审计查看) 权限配置文件。拥有 "System Administrator" (系统管理员) 权限配置文件的管理员可以分析审计记录。

要配置角色以处理审计服务，请参见第 153 页中的“初次配置 RBAC (任务列表)”。

审计和 Oracle Solaris 区域

非全局区域可以像全局区域一样进行审计，可以设置自己的标志、存储和审计策略。

当以相同方式审计所有区域时，全局区域中的 `audit_class` 和 `audit_event` 文件会为每个区域中的审计提供类到事件的映射。对于按区域名称后选记录，`+zonename` 策略选项非常有用。

区域也可以分别审计。在全局区域设置策略选项 `perzone` 时，每个非全局区域都会运行自己的审计服务，处理自己的审计队列，指定其审计记录的内容和位置。一个非全局区域也可以设置多个审计策略选项。它不能设置影响整个系统的策略，所以非全局区域无法设置 `ahlt` 或 `perzone` 策略。有关更多说明，请参见第 487 页中的“在具有 Oracle Solaris 区域的系统上审计”和第 492 页中的“如何规划区域中的审计”。

要了解有关区域的信息，请参见《Oracle Solaris 管理：Oracle Solaris Zones、Oracle Solaris 10 Zones 和资源管理》中的第 II 部分，“Oracle Solaris Zones”。

审计类

Oracle Solaris 定义了作为容纳大量审计事件的方便容器的审计类。

您可以重新配置审计类，也可以创建新的审计类。审计类名称的长度最多为 8 个字符。类描述最多可包含 72 个字符。允许使用数字和非字母数字字符。有关更多信息，请参见 `audit_class(4)` 手册页和第 513 页中的“如何添加审计类”。



注意 `-all` 类可以生成大量数据并快速填满磁盘。仅当有特殊理由需审计所有活动时，才使用 `all` 类。

审计类语法

可以针对成功、失败或两者对审计类中的事件进行审计。

- 如果不带前缀，则同时针对成功和失败两种情况对事件类进行审计。
- 如果带有加号(+)前缀，则仅针对成功情况对事件类进行审计。
- 如果带有减号(-)前缀，则仅针对失败情况对事件类进行审计。
- 在前缀或审计标志之前使用插入记号(^)，可修改当前预选。例如，
 - 如果为系统预选了 `ot`，而用户的预选为 `^ot`，则不会针对 `other` 类中的事件审计该用户。
 - 如果为系统预选了 `+ot`，而用户的预选为 `^+ot`，则不会针对 `other` 类中的成功事件审计该用户。
 - 如果为系统预选了 `-ot`，而用户的预选为 `^-ot`，则不会针对 `other` 类中的失败事件审计该用户。

要查看审计类预选的语法，请参见 [audit_flags\(5\)](#) 手册页。

可以在以下命令中指定审计类及其前缀：

- 作为 `auditconfig` 命令选项 `-setflags` 和 `-setnaflags` 的参数。
- 作为 `audit_syslog` 插件的 `p_flags` 属性的值。您可以指定该属性作为 `auditconfig -setplugin audit_syslog active` 命令的一个选项。
- 作为 `useradd`、`usermod`、`roleadd` 和 `rolemod` 命令的 `-K audit_flags=always-audit-flags:never-audit-flags` 选项的值。
- 作为 `profiles` 命令的 `-always_audit` 和 `-never_audit` 属性的值。

审计插件

审计插件指定如何处理审计队列中的审计记录。可以按名称指定审计插件：`audit_binfile`、`audit_remote` 和 `audit_syslog`，可用作 `auditconfig -setplugin` 命令的参数。可通过以下属性进一步指定插件：

- `audit_binfile` 插件
 - 将二进制数据发送到的位置—`p_dir` 属性
 - 在向管理员发出警告之前，磁盘上剩余的最小空间量—`p_minfree` 属性
 - 审计文件的最大大小—`p_fsize` 属性
- `audit_remote` 插件
 - 要将二进制审计数据发送到的远程验证审计服务器—`p_hosts` 属性
 - 尝试访问远程验证审计服务器的次数—`p_retries` 属性
 - 两次尝试访问远程验证审计服务器之间的秒数—`p_timeout` 属性

- `audit_syslog` 插件
 - 选择要发送到 `syslog` 的审计记录的文本摘要 — `p_flags` 属性
- 对于所有插件，插件排队的审计记录的最大数量 — `qsize` 属性

请参阅 [audit_binfile\(5\)](#)、[audit_remote\(5\)](#)、[audit_syslog\(5\)](#) 和 [auditconfig\(1M\)](#) 手册页。

审计策略

审计策略可确定是否将其他信息添加到审计迹中。

以下策略向审计记录添加标

记：`arge`、`argv`、`group`、`path`、`seq`、`trail`、`windata_down`、`windata_up` 和 `zonename`。Oracle Solaris 的 Trusted Extensions 功能使用 `windata_down` 和 `windata_up` 策略。有关更多信息，请参见《[Trusted Extensions 配置和管理](#)》中的第 22 章“[Trusted Extensions 审计（概述）](#)”。

其余的策略则不添加标记。`public` 策略限制对公共文件的审计。`perzone` 策略针对非全局区域建立单独的审计队列。`ahlt` 和 `cnt` 策略确定无法传送审计记录时所发生的情况。有关详细信息，请参见第 558 页中的“[同步事件和异步事件的审计策略](#)”。

第 496 页中的“[了解审计策略](#)”中介绍了不同审计策略选项的影响。有关审计策略选项的说明，请参见 [auditconfig\(1M\)](#) 手册页中的 `-setpolicy` 选项。要获得可用策略选项的列表，请运行 `auditconfig -lspolicy` 命令。要获得当前策略，请运行 `auditconfig -getpolicy` 命令。

同步事件和异步事件的审计策略

`ahlt` 策略和 `cnt` 策略一起控制由于审计队列已满而无法接受更多事件时发生的状况。

注 – 如果至少一个插件的队列可以接受审计记录，则不会触发 `cnt` 或 `ahlt` 策略。

`cnt` 和 `ahlt` 策略独立且相关。结合使用这些策略可以带来以下效果：

- `-ahlt +cnt` 是随附的缺省策略。使用该缺省策略，即使无法记录审计的事件，也可以对它进行处理。
 - `-ahlt` 策略指明，如果无法将一个异步事件的审计记录放入内核审计队列，系统将对事件计数并继续处理。在全局区域，`as_dropped` 计数器会记录该计数。
 - `+cnt` 策略指明，如果一个同步事件已到达却无法被放入内核审计队列，系统将对事件计数并继续处理。区域的 `as_dropped` 计数器会记录该计数。
 - `-ahlt +cnt` 配置通常在那些即使继续处理会导致审计记录丢失，也必须继续处理的站点上使用。`auditstatdrop` 字段显示区域中丢弃的审计记录数。

- **+ahlt -cnt** 策略指明，当异步事件无法添加到内核审计队列时，处理将会停止。
+ahlt 策略指明，如果一个异步事件的审计记录无法被放入内核审计队列，所有处理都将停止。系统将进入混乱状态。异步事件将不会进入审计队列，而必须通过调用栈上的指针恢复。
-cnt 策略指明，如果一个同步事件无法被放入内核审计队列，则会阻塞尝试发送事件的线程。该线程将被放入休眠队列，直到审计空间可用。不会保留计数。在审计空间可用之前，程序看上去可能处于挂起状态。
+ahlt -cnt 配置通常在每个审计事件记录的重要性高于系统可用性的站点上使用。在审计空间可用之前，程序看上去处于挂起状态。`auditstat wblk` 字段显示线程被阻塞的次数。
 但是，如果发生异步事件，系统将进入混乱状态，从而导致故障。可以通过保存的故障转储手动恢复审计事件的内核队列。异步事件将不会进入审计队列，而必须通过调用栈上的指针恢复。
- **-ahlt -cnt** 策略指明，如果异步事件无法被放入内核审计队列，将对事件计数，并继续处理。如果一个同步事件无法被放入内核审计队列，则会阻塞尝试发送事件的线程。该线程将被放入休眠队列，直到审计空间可用。不会保留计数。在审计空间可用之前，程序看上去可能处于挂起状态。
-ahlt -cnt 配置通常在那些所有同步审计事件的记录比部分异步审计记录的潜在损失重要的站点上使用。`auditstat wblk` 字段显示线程被阻塞的次数。
- **+ahlt +cnt** 策略指明，如果异步事件无法放入内核审计队列，系统将进入混乱状态。如果无法将同步事件放入内核审计队列，系统将对事件计数，然后继续处理。

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**—系统范围审计掩码和用户特定审计掩码的组合（如果指定了用户审计掩码）。用户登录时，登录过程将合并预选类，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：

```
(system-wide default flags + always-audit-classes) - never-audit-classes
```

将 `auditconfig -getflags` 命令结果中的系统范围审计类添加到用户的 `always_audit` 关键字值 `always-audit-classes` 中的类。然后，从总数中减去用户的 `never-audit-classes` 中的类。另请参见 [audit_flags\(5\)](#) 手册页。

- **审计用户 ID**—用户登录时，进程便会获取不变的审计用户 ID。由用户初始进程启动的所有子进程都会继承此 ID。审计用户 ID 有助于履行职责。即使在用户承担角色后，审计用户 ID 仍保持不变。保存在每个审计记录中的审计用户 ID 使您能够始终跟踪追溯到登录用户的操作。
- **审计会话 ID**—审计会话 ID 在登录时指定。此 ID 将由所有子进程继承。

- **终端 ID**—对于本地登录，终端 ID 包含本地系统的 IP 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编号为 0,0。对于远程登录，终端 ID 包含远程主机的 IP 地址，后跟远程端口号和本地端口号。

审计迹

审计迹包含二进制审计文件。该迹由 `audit_binfile` 插件创建。审计服务收集审计迹记录，并将它们发送到插件，通过插件将它们写入磁盘。

二进制审计文件名称约定

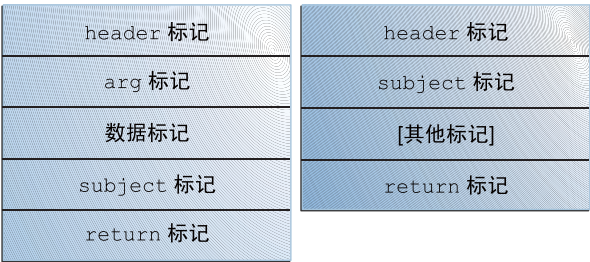
`audit_binfile` 插件可创建二进制审计文件。每个二进制审计文件都是自包含的记录集合。文件的名称标识生成记录的时间跨度以及生成这些记录的系统。指示时间范围的时间戳是以国际协调时间 (Coordinated Universal Time, UTC) 指定的，以确保它们即使跨越时区也能够以正确的顺序排列。

有关更多信息，请参见 `audit.log(4)` 手册页。有关打开和关闭的审计文件名称的示例，请参见第 537 页中的“如何清除 `not_terminated` 审计文件”。

审计记录结构

审计记录是一系列的审计标记。每个审计标记都包含用户 ID、时间和日期等事件信息。审计记录以 `header` 标记，以可选的 `trailer` 标记结束。其他审计标记包含与审计事件相关的信息。下图显示了典型内核审计记录和典型用户级审计记录。

图 29-1 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迹中后选记录。可以使用两种方法之一来解析收集的二进制数据。

- 您可以使用 `praudit` 命令。该命令的选项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解析二进制数据的字段。请注意，无法保证 Oracle 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 535 页中的“如何查看二进制审计文件的内容”。
有关每个审计标记的 `praudit` 输出的示例，请参见第 561 页中的“审计标记格式”中的各个标记。
- 您可以编写程序来解析二进制数据流。此程序必须考虑审计记录的变体。例如，`ioctl()` 系统调用会创建 "Bad file name"（错误的文件名称）的审计记录。该记录包含 "Invalid file descriptor"（无效的文件说明符）的 `ioctl()` 审计记录的不同标记。
 - 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。
 - 有关清单值，请参见 `/usr/include/bsm/audit.h` 文件。
 - 要查看审计记录中的标记顺序，请使用 `auditrecord` 命令。`auditrecord` 命令的输出包括不同清单值的不同标记。方括号 ([]) 表明，审计标记是可选的。有关更多信息，请参见 `auditrecord(1M)` 手册页。

审计标记格式

每个审计标记都有一个标记类型标识符，后跟标记的特定数据。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 29-1 用于审计的审计标记

标记名称	说明	更多信息
<code>acl</code>	访问控制条目 (Access Control Entry, ACE) 和访问控制列表 (Access Control List, ACL) 信息	第 563 页中的“ <code>acl</code> 标记”
<code>arbitrary</code>	具有格式和类型信息的数据	请参见 <code>audit.log(4)</code> 手册页。
<code>argument</code>	系统调用参数值	第 563 页中的“ <code>argument</code> 标记”
<code>attribute</code>	文件 <code>vnode</code> 信息	第 563 页中的“ <code>attribute</code> 标记”
<code>cmd</code>	命令参数和环境变量	第 563 页中的“ <code>cmd</code> 标记”
<code>exec_args</code>	可执行的系统调用参数	第 564 页中的“ <code>exec_args</code> 标记”
<code>exec_env</code>	可执行的系统调用环境变量	第 564 页中的“ <code>exec_env</code> 标记”
<code>exit</code>	程序退出信息	请参见 <code>audit.log(4)</code> 手册页。
<code>file</code>	审计文件信息	第 564 页中的“ <code>file</code> 标记”
<code>fmri</code>	框架管理资源指示器	第 565 页中的“ <code>fmri</code> 标记”

表 29-1 用于审计的审计标记 (续)

标记名称	说明	更多信息
group	进程组信息	第 565 页中的“group 标记”
header	表明审计记录开始	第 565 页中的“header 标记”
ip	IP 数据包头信息	请参见 <code>audit.log(4)</code> 手册页。
ip address	Internet 地址	第 566 页中的“ip address 标记”
ip port	Internet 端口地址	第 566 页中的“ip port 标记”
ipc	System V IPC 信息	第 566 页中的“ipc 标记”
IPC_perm	System V IPC 对象访问信息	第 567 页中的“IPC_perm 标记”
opaque	非结构化数据（未指定的格式）	请参见 <code>audit.log(4)</code> 手册页。
path	路径信息	第 567 页中的“path 标记”
path_attr	访问路径信息	第 567 页中的“path_attr 标记”
privilege	特权集信息	第 567 页中的“privilege 标记”
process	进程信息	第 568 页中的“process 标记”
return	系统调用的状态	第 568 页中的“return 标记”
sequence	序列号	第 568 页中的“sequence 标记”
socket	套接字类型和地址	第 568 页中的“socket 标记”
subject	主题信息（格式与 process 相同）	第 569 页中的“subject 标记”
text	ASCII 字符串	第 569 页中的“text 标记”
trailer	指示审计记录的结束	第 569 页中的“trailer 标记”
use of authorization	授权使用情况	第 570 页中的“use of authorization 标记”
use of privilege	特权使用情况	第 570 页中的“use of privilege 标记”
user	用户 ID 和用户名	第 570 页中的“user 标记”
xclient	X 客户机标识	第 570 页中的“xclient 标记”
zonename	区域名称	第 570 页中的“zonename 标记”
Trusted Extensions 标记	label 和 X 窗口系统信息	请参见《Trusted Extensions 配置和管理》中的“Trusted Extensions 审计参考”。

以下标记已过时：

- liaison
- host

■ tid

有关过时标记的信息，请参见包括标记的发行版参考资料。

审计记录始终以 **header** 标记开始。**header** 标记指示审计记录在审计迹中的开始位置。对于可归属事件，**subject** 和 **process** 标记引用导致事件发生的进程的值。对于无归属事件，**process** 标记引用系统。

acl 标记

acl 标记采用两种形式来记录有关适用于 ZFS 文件系统的访问控制条目 (Access Control Entry, ACE) 和适用于 UFS 文件系统的访问控制列表 (Access Control List, ACL) 的信息。

记录 UFS 文件系统的 **acl** 标记时，**praudit -x** 命令将显示如下字段：

```
<acl type="1" value="root" mode="6"/>
```

记录 ZFS 数据集的 **acl** 标记时，**praudit -x** 命令将显示如下字段：

```
<acl who="root" access_mask="default" flags="-i,-R" type="2"/>
```

argument 标记

argument 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。该标记允许在审计记录中使用 32 位整数的系统调用参数。

praudit -x 命令按如下方式显示 **argument** 标记的字段：

```
<argument arg-num="2" value="0x5401" desc="cmd"/>
```

attribute 标记

attribute 标记包含文件 **vnode** 中的信息。

attribute 标记通常与 **path** 标记同时出现。**attribute** 标记是在搜索路径期间生成的。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** 标记不包括在审计记录中。**praudit -x** 命令按如下方式显示 **attribute** 标记的字段：

```
<attribute mode="20620" uid="root" gid="tty" fsid="0" nodeid="9267" device="108233"/>
```

cmd 标记

cmd 标记记录与命令关联的参数列表和环境变量的列表。

`praudit -x` 命令显示 `cmd` 标记的字段：下面是一个截断的 `cmd` 标记。由于显示的原因进行了换行。

```
<cmd><arg>WINDOWID=6823679</arg>
<arg>COLORTERM=gnome-terminal</arg>
<arg>...LANG=C</arg>...<arg>HOST=machine1</arg>
<arg>LPDEST=printer1</arg>...</cmd>
```

exec_args 标记

`exec_args` 标记记录 `exec()` 系统调用的参数。

`praudit -x` 命令按如下方式显示 `exec_args` 标记的字段：

```
<exec_args><arg>/usr/bin/sh</arg><arg>/usr/bin/hostname</arg></exec_args>
```

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` 标记记录 `exec()` 系统调用的当前环境变量。

`praudit -x` 命令显示 `exec_env` 标记的字段。由于显示的原因进行了换行。

```
<exec_env><env>_=/usr/bin/hostname</env>
<env>LANG=C</env><env>PATH=/usr/bin:/usr/ucb</env>
<env><env>LOGNAME=jdoe</env><env>USER=jdoe</env>
<env>DISPLAY=:0</env><env>SHELL=/bin/csh</env>
<env>HOME=/home/jdoe</env><env>PWD=/home/jdoe</env><env>TZ=US/Pacific</env>
</exec_env>
```

注 – 仅当 `argv` 审计策略选项处于活动状态时，才输出 `exec_env` 标记。

file 标记

`file` 标记是一种特殊标记，它可在停用旧文件时标记新审计文件的开始以及旧审计文件的结束。初始 `file` 标记识别审计迹中的上一个文件。最终 `file` 标记识别审计迹中的下一个文件。这些标记一起将连续的审计文件“链接”成一个审计迹。

`praudit -x` 命令显示 `file` 标记的字段。由于显示的原因进行了换行。

```
<file iso8601="2009-04-08 14:18:26.200 -07:00">
/var/audit/machine1/files/20090408211826.not_terminated.machine1</file>
```


fmri 标记

fmri 标记记录了故障管理资源指示器 (fault management resource indicator, FMRI) 的用法。有关更多信息，请参见 [smf\(5\)](#) 手册页。

praudit -x 命令显示 fmri 标记的内容：

```
<fmri service_instance="svc:/system/cryptosvc"></fmri>
```

group 标记

group 标记记录进程凭证中的组项。

praudit -x 命令按如下方式显示 groups 标记的字段：

```
<group><gid>staff</gid><gid>other</gid></group>
```

注 - 仅当 group 审计策略选项处于活动状态时，才输出 group 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，将记录中的所有其他标记括在一起。

在极少的情况下，header 标记可能会包含一个或多个事件修饰符：

- fe 指示失败的审计事件
- fp 指示对特权的使用失败
- na 表示无归属事件

```
header,52,2,system booted,na,mach1,2011-10-10 10:10:20.564 -07:00
```

- rd 表示从对象中读取数据
- sp 表明特权的使用成功

```
header,120,2,exit(2),sp,mach1,2011-10-10 10:10:10.853 -07:00
```

- wr 表示向对象写入数据

praudit 命令按如下方式显示 header 标记：

```
header,756,2,execve(2),,machine1,2010-10-10 12:11:10.209 -07:00
```

praudit -x 命令在审计记录开始显示 header 标记的字段。由于显示的原因进行了换行。

```
<record version="2" event="execve(2)" host="machine1"
iso8601="2010-10-10 12:11:10.209 -07:00">
```

ip address 标记

ip address 标记包含 Internet 协议地址（IP 地址）。IP 地址以 IPv4 或 IPv6 格式显示。IPv4 地址使用 4 个字节。IPv6 地址使用 1 个字节来表示地址类型，使用 16 个字节来表示地址。

praudit -x 命令按如下方式显示 ip address 标记的内容：

```
<ip_address>machine1</ip_address>
```

ip port 标记

ip port 标记包含 TCP 或 UDP 端口地址。

praudit 命令按如下方式显示 ip port 标记：

```
ip port,0xf6d6
```

ipc 标记

ipc 标记包含调用者用于标识特殊 IPC 对象的 System V IPC 消息句柄、信号句柄或共享内存句柄。

注 – IPC 对象标识符不符合审计标记的上下文无关性质。没有可唯一标识 IPC 对象的全局名称。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，IPC 对象的标识应该不存在问题。很少用到 System 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 29-2 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 -x` 命令按如下方式显示 `ipc` 标记的字段：

```
<IPC ipc-type="shm" ipc-id="15"/>
```

IPC_perm 标记

`IPC_perm` 标记包含 System V IPC 访问权限的副本。该标记将被添加到由 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。

`praudit -x` 命令显示 `IPC_perm` 标记的字段。由于显示的原因进行了换行。

```
<IPC_perm uid="jdoe" gid="staff" creator-uid="jdoe"
creator-gid="staff" mode="100600" seq="0" key="0x0"/>
```

从与 IPC 对象关联的 `IPC_perm` 结构取得这些值。

path 标记

`path` 标记包含对象的访问路径信息。

`praudit -x` 命令显示 `path` 标记的内容：

```
<path>/export/home/srv/.xsession-errors</path>
```

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` 标记对象下的属性文件对象的顺序。系统调用存取属性文件，如 `openat()`。有关属性文件对象的更多信息，请参见 [fsattr\(5\)](#) 手册页。

`praudit` 命令显示 `path_attr` 标记如下：

```
path_attr,1,attr_file_name
```

privilege 标记

`privilege` 标记记录进程中的特权使用情况。不对基本集中的特权记录 `privilege` 标记。如果已通过管理操作从基本集中删除了特权，则会记录该特权的使用。有关特权的更多信息，请参见第 138 页中的“[特权（概述）](#)”

`praudit -x` 命令显示 `privilege` 标记的字段。

```
<privilege set-type="Inheritable">ALL</privilege>
```

process 标记

process 标记包含有关与进程关联的用户（如信号接收者）的信息。

praudit -x 命令显示 process 标记的字段。由于显示的原因进行了换行。

```
<process audit-uid="-2" uid="root" gid="root" ruid="root"
rgid="root" pid="567" sid="0" tid="0 0 0.0.0.0"/>
```

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)。

return 标记始终作为系统调用的内核生成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 命令按如下方式显示 return 标记的字段：

```
<return errval="failure: Operation now in progress" retval="-1"/>
```

sequence 标记

sequence 标记包含一个序列号。每次向审计迹添加一条审计记录，序列号就增加一个数字。该标记用于调试。

praudit -x 命令显示 sequence 标记的内容：

```
<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 标记包含描述 Internet 套接字的信息。在某些实例中，标记仅包括远程端口和远程 IP 地址。

praudit 命令按如下方式显示该 socket 标记实例：

```
socket,0x0002,0x83b1,localhost
```

展开的标记可添加信息，包括套接字类型和本地端口信息。

`praudit -x` 命令按如下方式显示该 `socket` 标记实例。由于显示的原因进行了换行。

```
<socket sock_domain="0x0002" sock_type="0x0002" lport="0x83cf"
laddr="example1" fport="0x2383" faddr="server1.Subdomain.Domain.COM"/>
```

subject 标记

`subject` 标记描述执行或尝试执行某项操作的用户。格式与 `process` 标记的格式相同。

`subject` 标记始终作为系统调用的内核生成审计记录的一部分返回。`praudit` 命令按如下方式显示 `subject` 标记：

```
subject,jdoe,root,root,root,root,1631,1421584480,8243 65558 machine1
```

`praudit -x` 命令显示 `subject` 标记的字段。由于显示的原因进行了换行。

```
<subject audit-uid="jdoe" uid="root" gid="root" ruid="root"
rgid="root" pid="1631" sid="1421584480" tid="8243 65558 machine1"/>
```

text 标记

`text` 标记包含一个文本字符串。

`praudit -x` 命令显示 `text` 标记的内容：

```
<text>booting kernel</text>
```

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` 标记指示审计记录的开头。`trailer` 标记指示审计记录的结尾。`trailer` 标记是可选标记。仅当已设置 `trail` 审计策略选项时，才能将 `trailer` 标记添加为每条记录的最后一个标记。

在 `trailer` 打开的情况下生成审计记录时，`auditreduce` 命令可以验证 `trailer` 是否正确指回记录 `header`。`trailer` 标记支持向后查找审计迹。

`praudit` 命令按如下方式显示 `trailer` 标记：

```
trailer,136
```

use of authorization 标记

use of authorization 标记记录授权使用。

praudit 命令按如下方式显示 use of authorization 标记：

```
use of authorization,solaris.role.delegate
```

```
XXXX<use_of_authorization result="successful use of auth">solaris.role.delegate</use_of_auth>
```

use of privilege 标记

use of privilege 标记记录特权使用。

praudit -x 命令按如下方式显示 use of privilege 标记的字段：

```
<use_of_privilege result="successful use of priv">proc_setid</use_of_privilege>
```

user 标记

user 标记记录用户名和用户 ID。如果用户名不同于调用者，则出现此标记。

praudit -x 命令按如下方式显示 user 标记的字段：

```
<user uid="123456" username="tester1"/>
```

xclient 标记

xclient 标记包含到 X 服务器的客户机连接数目。

praudit -x 命令按如下方式显示 xclient 标记的内容：

```
<X_client>15</X_client>
```

zonename 标记

zonename 标记记录发生审计事件的区域。字符串 "global" 指示审计事件发生在全局区域。

praudit -x 命令显示 zonename 标记的内容：

```
<zone name="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，通过 ACL 可以让组获得对某个文件的读取权限，而仅允许该组中的一个成员获得对该文件的写入权限。
admin principal (管理主体)	名称形式为 <code>username/admin</code> 的用户主体（如 <code>jdoe/admin</code> ）。与一般用户主体相比，管理主体可以拥有更多特权（例如，可以更改策略）。另请参见 principal name (主体名称) 和 user principal (用户主体) 。
AES	高级加密标准 (Advanced Encryption Standard)。一种对称的 128 位块数据加密技术。美国政府在 2000 年 10 月采用该种算法的 Rijndael 变体作为其加密标准。AES 从而取代了 user principal (用户主体) 加密方法成为政府的加密标准。
algorithm (算法)	加密算法。这是一种确立的递归计算过程，用于对输入执行加密或散列操作。
application server (应用服务器)	请参见 network application server (网络应用服务器) 。
asynchronous audit event (异步审计事件)	异步事件在系统事件中属于少数。这些事件不与任何进程关联，因此没有任何进程可供阻塞并在以后唤醒。例如，初始系统引导和 PROM 进入和退出事件就属于异步事件。
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文件系统中。
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迹所包括的可选信息。 <code>cnt</code> 和 <code>ahlt</code> 这两个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 (审计迹)	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，以及时间戳。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 set (基本特权集)	登录时为用户进程指定的特权集。在未修改的系统上，每个用户的初始可继承特权集等同于登录时获取的基本特权集。
Blowfish	一种对称块加密算法，它采用 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 保护的网路服务的客户机（用户或应用程序）。客户机主体名称将以 <code>rpc_gss_principal_t</code> 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路服务的进程，例如，使用 <code>rlogin</code> 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 <code>krb5.conf</code> 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 Oracle Solaris 的加密框架功能中，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。例如，Kerberos、IKE 和 IPsec 便属于使用者。有关提供者的示例，请参见 provider (提供者) 。
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) 和 session key (会话密钥) 。
cryptographic algorithm (密码算法)	请参见 algorithm (算法) 。
DES	Data Encryption Standard（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	内核级别的设备保护。设备策略在设备上作为两个特权集实现。一个特权集控制对设备的读取权限，另一个特权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，两个用户可以在以前没有任何密钥的情况下通过不安全的介质交换密钥。Diffie-Hellman 由 Kerberos 使用。
digest (摘要)	请参见 message digest (消息摘要) 。
DSA	Digital Signature Algorithm（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 (有效特权集)	当前对进程有效的特权集。
flavor (特性)	以前, <i>security flavor</i> (安全特性) 和 <i>authentication flavor</i> (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 也是一种安全特性, 虽然它除了验证之外还提供完整性和保密性服务。
forwardable ticket (可转发票证)	一种票证, 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。例如, 如果用户 david 从用户 jennifer 的计算机上获取了可转发票证, 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	Fully qualified domain name (全限定域名)。例如, central.example.com (与简单的 denver 相对)。
GSS-API	Generic Security Service Application Programming Interface (通用安全服务应用编程接口)。为各种模块化安全服务 (包括 Kerberos 服务) 提供支持的网路层。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 (硬件提供者)	在 Oracle Solaris 的加密框架功能中, 是指设备驱动程序及其硬件加速器。硬件提供者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 (提供者)。
host	可通过网络进行访问的系统。
host principal (主机主体)	服务主体的一个特定实例, 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。例如, host/central.example.com@EXAMPLE.COM 便是一个主机主体。另请参见 server principal (服务器主体)。
inheritable set (可继承特权集)	进程可以通过调用 exec 而继承的特权集。
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 <i>initial</i> , 以便使其自身确信客户机知晓其密钥。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)	尚未成为可用票证的以后生效的票证。应用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	Key Distribution Center（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。 Oracle 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合。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）。
key	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，但由服务器和服务使用。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时间。另请参见 session key（会话密钥）。
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大致相同。
kvno	Key version number（密钥版本号）。按照生成顺序跟踪特定密钥的序列号。kvno 最高则表示密钥最新。
least privilege（最小特权）	一种安全模型，该模型仅向指定进程提供超级用户功能的某个子集。最小特权模型为一般用户指定可以用来执行个人管理任务（如挂载文件系统和更改文件的所有权）的足够特权。另一方面，仅使用完成该任务所需的特权运行进程，而不是使用超级用户的完全功能模式（即所有特权）。对非超级用户而言，可以包含由于编程错误而导致的损坏（如缓冲区溢出），该用户对重要功能（如读取或写入受保护的系统文件或停止计算机）没有访问权限。
limit set（限制特权集）	对哪些特权可用于进程及其子进程的外部限制。

MAC	1. 请参见 message authentication code, MAC（消息验证码）。 2. 也称为标签设置操作。在政府安全术语中，MAC 是指 Mandatory Access Control（强制访问控制）。例如，Top Secret（绝密）和 Confidential（机密）之类的标签便是 MAC。MAC 与 DAC 相对，后者是指 Discretionary Access Control（自主访问控制）。例如，UNIX 权限便是一个 DAC。 3. 在硬件中，是指 LAN 中的唯一系统地址。如果系统位于以太网中，则 MAC 是指以太网地址。
master KDC（主 KDC）	每个领域中的主要 KDC，包括 Kerberos 管理服务器 <code>kadmind</code> ，以及验证和票证授予守护进程 <code>krb5kdc</code> 。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 Oracle Solaris 的加密框架功能中，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。不安装与服务器操作不直接相关的任何软件，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，由指定的命名服务（如 NIS 或 LDAP）提供服务的单个主机或所有主机。
network application server（网络应用服务器）	提供网络应用的服务器，如 <code>ftp</code> 。一个领域可以包含多个网络应用服务器。
nonattributable audit event（无归属审计事件）	无法确定其触发者的审计事件，如 <code>AUE_BOOT</code> 事件。
NTP	Network Time Protocol（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	Pluggable Authentication Module（可插拔验证模块）。一种框架，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。PAM 可用于在登录时初始化 Kerberos 会话。

passphrase (口令短语)	一种短语，用于验证某个私钥是否是由口令短语用户创建。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进行验证。
password policy	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，允许的错误输入次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 (允许特权集)	可供进程使用的特权集。
policy for public key technologies (公钥技术的策略)	在密钥管理框架 (Key Management Framework, KMF) 中，所实现的策略是管理证书的使用。KMF 策略数据库可以对由 KMF 库管理的密钥和证书的使用施加约束。
policy in the Cryptographic Framework	在 Oracle Solaris 的加密框架功能中，所实现的策略是禁用现有的加密机制。从而使这些机制不可使用。加密框架中的策略可能会阻止使用提供者（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合和相关措施，可用于定义所处理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，使用相应特权保护设备，以及每六周更改一次口令。 有关 Oracle 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、device policy、Kerberos policy (Kerberos 策略)、password policy和RBAC policy。
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，因为在运行批处理作业之前无法使用该票证（即使被盗）。颁发以后生效的票证时，将以 <i>invalid</i> 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) 客户机请求 KDC 进行验证。通常，以后生效的票证在票证授予票证的截止时间之前会一直有效。但是，如果将以后生效的票证标记为 <i>renewable</i> ，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	1. 参与网络通信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。换言之，主体是 Kerberos 可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)。
principal name (主体名称)	1. 主体的名称，格式为 <i>primary/instance@REALM</i> 。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)。
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)	采用私钥加密时, 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 。
privilege	Oracle Solaris 系统中的进程具有的独立权限。与 root 相比, 特权可提供更为精细的进程控制。特权是在内核中定义和实施的。有关特权的完整说明, 请参见 privileges(5) 手册页。
privilege-aware (特权识别)	在其代码中启用和禁用特权的程序、脚本和命令。在生产环境中, 启用的特权必须提供给进程, 例如, 通过要求程序的用户使用将特权添加到程序中的权限配置文件。有关特权的完整说明, 请参见 privileges(5) 手册页。
privilege escalation (特权升级)	获取对指定给您的安全属性所属资源范围之外的资源的访问权, 包括覆盖和允许。特权升级的结果是某个进程可以执行未经授权的操作。
privilege model (特权模型)	计算机系统上比超级用户模型更为严格的安全模型。在特权模型中, 进程需要具有相应的特权才能运行。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特权。可以将特权指定给管理员的登录过程。或者, 可以指定特权只对特定命令有效。
privilege set (特权集)	特权的集合。每个进程都有四个特权集, 用于确定进程是否可以使用特定特权。请参见 limit set (限制特权集)、 effective set (有效特权集)、 permitted set (允许特权集) 和 inheritable set (可继承特权集)。
	此外, 特权的 basic set (基本特权集) 是指登录时为用户进程指定的特权集合。
privileged application (特权应用程序)	可以覆盖系统控制的应用程序。该应用程序可以检查安全属性 (如特定的 UID、GID、授权或特权)。
profile shell (配置文件 shell)	在 RBAC 中, 角色 (或用户) 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 (提供者)	在 Oracle Solaris 的加密框架功能中, 是指为用户提供者的加密服务。例如, PKCS #11 库、内核加密模块和硬件加速器便是提供者。提供者可插入到加密框架中, 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。采用公钥加密时，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) 。
public object (公共对象)	root 用户所拥有且全局可读的文件，如 /etc 目录中的任何文件。
QOP	Quality of Protection (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，Oracle Solaris 的一项功能。对超级用户模型（管理员对系统要么具有全部控制权要么毫无控制权）的替代。使用 RBAC，组织可以将超级用户的功能分离开来，并将各功能指定给称为角色的特殊用户帐户。可以根据个体的职责将角色指定给特定的个体。
RBAC policy	与命令关联的安全策略。目前，solaris 是有效的策略。solaris 策略可识别特权、授权和 setuid 安全属性。
realm (领域)	- 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。 - 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 <i>renewable</i> 。可更新票证有两个截止时间：a) 票证的当前实例的截止时间，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，则可在首次失效之前更新此票证。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。当某个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、特权、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，仅指定用户才能承担此身份。
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。
scan engine (扫描引擎)	第三方应用程序，驻留在外部主机上，可检查文件中是否含有已知病毒。
SEAM	Sun Enterprise Authentication Mechanism (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 (密钥)	请参见 private key (私钥) 。
Secure Shell (安全 Shell)	一种特殊协议，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。

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。在特权模型中, 安全属性是指特权。为某个命令提供特权后, 此命令便会成功执行。特权模型与超级用户模型兼容, 因为特权模型也可将 setuid 和 setgid 程序识别为安全属性。
security flavor (安全特性)	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 。
separation of duty (职责分离)	least privilege (最小特权) 的部分概念。职责分离可阻止一个用户执行或批准完成事务的所有操作。例如, 在 RBAC 中, 可以将登录用户的创建与安全覆盖的指定分隔开来。一个角色创建该用户。另一个角色可以将安全属性 (如权限配置文件、角色和特权) 指定给现有用户。
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 ssh 远程登录到系统 central.example.com , 则该系统便是提供 ssh 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 。
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)	- 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, 则该计算机便是提供 rlogin 服务的服务器。 - 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。生成会话密钥的目的是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时间。另请参见 key 。
SHA1	Secure Hashing Algorithm (安全散列算法)。该算法可以在长度小于 2^{64} 的任何输入上运行以生成消息摘要。 SHA1 算法是 DSA 的输入。

single-system image (单系统映像)	单系统映像用在 Oracle Solaris 审计中来描述使用相同命名服务的一组受审计系统。这些系统将其审计记录发送给某个中心审计服务器，可在该服务器中对记录进行比较，就像这些记录来自一个系统一样。
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个从 KDC (但仅有一个主 KDC)。另请参见 KDC 和 master KDC (主 KDC) 。
software provider (软件提供者)	在 Oracle Solaris 的加密框架功能中，是指提供加密服务的内核软件模块或 PKCS #11 库。另请参见 provider (提供者) 。
stash file (存储文件)	存储文件包含 KDC 主密钥的已加密副本。重新引导服务器时将使用此主密钥自动验证 KDC，之后 KDC 可启动 <code>kadmind</code> 和 <code>krb5kdc</code> 进程。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，管理员对系统要么具有全部的控制权要么毫无控制权。通常，为了管理计算机，用户可成为超级用户 (<code>root</code>)，并可执行所有管理活动。
synchronous audit event (同步审计事件)	审计事件中的大多数事件属于同步审计事件。这些事件与系统中的某个进程关联。与某个进程关联的无归属事件属于同步事件，如失败的登录。
TGS	Ticket-Granting Service (票证授予服务)。负责颁发票证的那部分 KDC。
TGT	Ticket-Granting Ticket (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、时间戳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)	属于某个特定用户的主体。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 <code>jdoe</code> 或 <code>jdoe/admin</code>)。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

[] (方括号), `auditrecord` 输出, 561
\$\$ (双美元符号), 父 `shell` 进程号, 180
> (重定向输出), 防止, 43
>> (附加输出), 防止, 43
"Audit Configuration" (审计配置) 权限配置文件, 555
"Audit Control" (审计控制) 权限配置文件, 555
"Audit Review" (审计查看) 权限配置文件, 556
~/.gkadmin 文件, 说明, 461
~/.k5login 文件, 说明, 461
~/.rhosts 文件, 说明, 296
~/.shosts 文件, 说明, 296
~/.ssh/authorized_keys 文件
 覆盖, 297
 说明, 295
~/.ssh/config 文件
 覆盖, 296
 说明, 296
~/.ssh/environment 文件, 说明, 296
~/.ssh/id_dsa 文件, 覆盖, 297
~/.ssh/id_rsa 文件, 覆盖, 297
~/.ssh/identity 文件, 覆盖, 297
~/.ssh/known_hosts 文件
 覆盖, 297
 说明, 296
~/.ssh/rc 文件, 说明, 296
@ (at 符号), `device_allocate` 文件, 89
^ (插入记号), 审计类前缀修饰符, 557
= (等号), 文件权限符号, 114
.(点)
 授权名称分隔符, 188

.(点) (续)
 显示隐藏文件, 117
\ (反斜杠)
 `device_allocate` 文件, 89
 `device_maps` 文件, 88
;(分号), `device_allocate` 文件, 88
+ (加号)
 `sulog` 文件, 62
 审计类前缀, 557
 文件权限符号, 114
- (减号)
 `sulog` 文件, 62
 审计类前缀, 557
 文件类型符号, 110
 文件权限符号, 114
(井号)
 `device_allocate` 文件, 89
 `device_maps` 文件, 88
* (星号)
 `device_allocate` 文件, 88, 89
 检查 RBAC 授权, 162
 通配符
 在 RBAC 授权中, 188
32 位可执行文件, 防止危及安全, 116
3des-cbc 加密算法, `ssh_config` 文件, 290
3des 加密算法, `ssh_config` 文件, 290

A

-A 选项, `auditreduce` 命令, 533

-a 选项

- auditrecord 命令, 530
- digest 命令, 212
- encrypt 命令, 215
- mac 命令, 213
- 基于 Kerberos 的命令, 456

ACL

- kadm5.acl 文件, 422, 424, 428
- 关于复制项的限制, 115
- 说明, 46, 115–116
- 项格式, 115–116

acl 审计标记, 格式, 563

add_drv 命令, 说明, 85

admin_server 部分

- krb5.conf 文件, 326, 332

AES 内核提供者, 219

aes128-cbc 加密算法, ssh_config 文件, 290

aes128-ctr 加密算法, ssh_config 文件, 290

ahlt 审计策略

- 设置, 510
- 使用 cnt 策略, 558–559
- 说明, 496

all 审计类, 使用注意事项, 556

allocate 命令

- 磁带机, 82
- 分配错误状态, 87
- 使用, 81–82
- 所需授权, 87
- 要求授权, 193
- 用户授权, 78

AllowGroups 关键字, sshd_config 文件, 290

AllowTcpForwarding 关键字

- sshd_config 文件, 290
- 更改, 275

AllowUsers 关键字, sshd_config 文件, 290

All (所有) (RBAC), 权限配置文件, 186

always-audit 类, 进程预选掩码, 559

arcfour 加密算法, ssh_config 文件, 290

ARCFOUR 内核提供者, 219

arge 审计策略

- 和 exec_env 标记, 564
- 设置, 545
- 说明, 496

argument 审计标记, 格式, 563

argv 审计策略

- 和 exec_args 标记, 564
- 设置, 545
- 说明, 497

@ 符号 (@), device_allocate 文件, 89

at 命令, 要求授权, 193

atq 命令, 要求授权, 193

attribute 审计标记, 563

audit -s 命令, 526–528, 529

audit -t 命令, 528–529

audit_binfile 插件, 482

获取属性, 519, 520

删除队列大小, 520

设置空闲空间警告, 520

设置属性, 518–520

限制审计文件大小, 519

audit_class 文件

故障排除, 513

添加类, 513

Audit Configuration (审计配置) 权限配置文件

配置审计策略, 508–511

审计角色, 159

显示审计缺省值, 503–504

预选审计类, 504–505

Audit Control (审计控制) 权限配置文件

禁用审计服务, 528–529

启用审计服务, 529

刷新审计服务, 526–528

audit_event 文件

安全删除事件, 548–549

更改类成员身份, 514

说明, 480

audit_flags 关键字, 504

使用, 557

使用插入记号 (^) 前缀, 506–507

指定审计预选的用户例外, 505–508

audit.notice 项, syslog.conf 文件, 522

audit_remote 插件, 482

获取属性, 520–521

设置属性, 520–521

audit_syslog 插件, 482

设置属性, 521–523

audit_warn 脚本

配置, 512–513

audit_warn脚本 (续)

说明, 555

audit命令

禁用审计服务, 528-529

刷新审计服务, 526-528

选项, 554

auditconfig命令

-getplugin选项, 520-521, 521-523

-setflags选项, 504-505

-setnaflags选项, 504-505

-setplugin选项, 520-521, 521-523

参数形式的审计类, 481

策略选项, 508-511

查看缺省审计预选, 504-505

队列控制选项, 511-512

配置策略, 508-511

配置队列控制, 511-512

设置audit_binfile属性, 518-520

设置audit_remote属性, 520-521

设置活动审计策略, 510

设置临时审计策略, 510

设置审计策略, 545

设置系统范围内的审计参数, 481

说明, 555

添加审计文件系统, 518-520

显示审计缺省值, 503-504

向远程系统信息库发送文件, 520-521

预选审计类, 504-505

auditd守护进程

刷新审计服务, 527, 528

auditlog文件, 文本审计记录, 522**auditrecord命令**

可选标记 ([I]), 561

列出程序的格式, 531

列出类的格式, 531-532

列出所有格式, 530

示例, 530

输出中的 [] (方括号), 561

说明, 555

显示审计记录定义, 530-532

auditreduce命令

-A选项, 533

-b选项, 534

-c选项, 533

auditreduce命令 (续)

-c选项, 535

-D选项, 533

-d选项, 535

-e选项, 534

-M选项, 533

-O选项, 532-533, 533, 534

trailer 标记, 和, 569

过滤选项, 533

合并审计记录, 532-533

清除审计文件, 537-538

时间戳使用, 532

使用大写选项, 532

使用小写选项, 533

示例, 532-533

说明, 555

选择审计记录, 533-535

auditstat命令, 说明, 555**auth_attr数据库**

说明, 190

摘要, 189

AUTH_DES验证, 请参见AUTH_DH验证

AUTH_DH验证, 和 NFS, 247

authlog文件, 保存失败的登录尝试, 57-59

authorized_keys文件, 说明, 295

AuthorizedKeysFile 关键字, sshd_config文件, 290

AUTHS_GRANTED 关键字, policy.conf文件, 191

auths命令, 说明, 192

auto_transition选项, SASL和, 266

auxprop_login选项, SASL和, 266

B

-b选项, auditreduce命令, 534

Banner 关键字, sshd_config文件, 290

BART

安全注意事项, 96

程序输出, 108

概述, 93-95

任务列表, 96-97

详细输出, 108

组件, 94-95

bart compare命令, 94

bart create命令, 94, 97

bart 命令, 93
 BART 中的引用语法, 107
 Basic Solaris User (基本 Solaris 用户) (RBAC), 权限配置文件, 186
 Batchmode 关键字, ssh_config 文件, 290
 BindAddress 关键字, ssh_config 文件, 290
 binding 控制标志, PAM, 261
 blowfish-cbc 加密算法, ssh_config 文件, 290
 Blowfish 加密算法
 policy.conf 文件, 60
 ssh_config 文件, 290
 Blowfish 加密算法, 内核提供者, 219
 Blowfish 加密算法
 允许在异构环境结构中, 60
 Bourne shell, 特权版本, 136

C

-C 选项, auditreduce 命令, 533
 C shell, 特权版本, 136
 -c 选项
 auditrecord 命令, 531–532
 auditreduce 命令, 535
 canon_user_plugin 选项, SASL 和, 266
 CD-ROM 驱动器
 安全性, 90
 分配, 83
 cdrw 命令, 要求授权, 193
 ChallengeResponseAuthentication 关键字, 请参见 KbdInteractiveAuthentication 关键字
 CheckHostIP 关键字, ssh_config 文件, 290
 chgrp 命令
 说明, 110
 语法, 119
 chkey 命令, 249, 253
 chmod 命令
 更改特殊权限, 121–122, 122
 说明, 110
 语法, 121
 chown 命令, 说明, 109
 ChrootDirectory 关键字, ssh_config 文件, 290
 Cipher 关键字, ssh_config 文件, 290
 Ciphers 关键字, 安全 Shell, 290
 clear 保护级别, 457

ClearAllForwardings 关键字, 安全 Shell 端口转发, 290
 ClientAliveCountMax 关键字, ssh_config 文件, 290
 ClientAliveInterval 关键字, ssh_config 文件, 290
 cIntntconfig 主体
 创建, 328, 334
 cmd 审计标记, 563–564
 cnt 审计策略
 使用 ahl1t 策略, 558–559
 说明, 497
 Compression 关键字, 安全 Shell, 290
 CompressionLevel 关键字, ssh_config 文件, 290
 ConnectionAttempts 关键字, ssh_config 文件, 290
 ConnectTimeout 关键字, ssh_config 文件, 290
 CONSOLE_USER 关键字, policy.conf 文件, 191
 Console User (控制台用户) (RBAC), 权限配置文件, 186
 crammd5.so.1 插件, SASL 和, 266
 cred 表
 DH 验证和, 248
 服务器存储的信息, 250
 cred 数据库, DH 验证, 248–251
 crontab 文件, 要求授权, 193
 CRYPT_ALGORITHMS_ALLOW 关键字, policy.conf 文件, 38
 CRYPT_ALGORITHMS_DEPRECATE 关键字, policy.conf 文件, 38
 crypt_bsdbf 口令算法, 38
 crypt_bsmd5 口令算法, 37
 CRYPT_DEFAULT 关键字, policy.conf 文件, 38
 CRYPT_DEFAULT 系统变量, 59
 crypt_sha256 口令算法, 38, 59–61
 crypt_sunmd5 口令算法, 38
 crypt_unix 口令算法, 38
 crypt 命令, 文件安全性, 46
 Crypto Management (加密管理) (RBAC), 创建角色, 158
 cryptoadm install 命令, 安装 PKCS #11 库, 223
 cryptoadm 命令
 -m 选项, 223, 225
 -p 选项, 224, 225
 安装 PKCS #11 库, 223
 恢复内核软件提供者, 225
 禁用加密机制, 223, 225

cryptoadm 命令 (续)

禁用硬件机制, 228-230

列出提供者, 219

说明, 202

Cryptoki, 请参见PKCS #11 库

csh 命令, 特权版本, 136

D**-D 选项**

auditreduce 命令, 533

ppriv 命令, 181

-d 选项

auditreduce 命令, 534, 535

dd 命令, 生成私钥, 206-208

deallocate 命令

分配错误状态, 87

设备清除脚本和, 90-91

使用, 84

所需授权, 87

要求授权, 193

decrypt 命令

说明, 203

语法, 216

default/login 文件, 说明, 296

default_realm 部分

krb5.conf 文件, 326, 332

delete_entry 命令, ktutil 命令, 445

DenyGroups 关键字, sshd_config 文件, 290

DenyUsers 关键字, sshd_config 文件, 290

DES 加密, 安全 NFS, 248

DES 加密, 内核提供者, 219

/dev/arp 设备, 获取 IP MIB-II 信息, 76

/dev/urandom 设备, 206-208

devfsadm 命令, 说明, 85

device_allocate 文件

格式, 88

说明, 88-89

样例, 80, 88

device_maps 文件

格式, 88

说明, 87

项样例, 88

DH 验证

共享文件, 254

挂载文件, 254

说明, 248-251

在 NIS 中配置, 252-253

针对 NIS 客户机, 252-253

Diffie-Hellman 验证, 请参见DH 验证

digest 命令

示例, 212

说明, 203

语法, 212

digestmd5.so.1 插件, SASL 和, 266

DisableBanner 关键字, ssh_config 文件, 290

dminfo 命令, 88

DNS, Kerberos 和, 315-316

domain_realm 部分

krb5.conf 文件, 315, 326, 332

DSAAuthentication 关键字, 请参

见PubkeyAuthentication 关键字

DynamicForward 关键字, ssh_config 文件, 290

E**-e 选项**

auditreduce 命令, 534

ppriv 命令, 181

ECC 内核提供者, 219

eeprom 命令, 36, 63-65

eject 命令, 设备清除和, 90

elfsign 命令, 说明, 203

encrypt 命令

错误消息, 217

故障排除, 217

说明, 203

语法, 206

EscapeChar 关键字, ssh_config 文件, 291

/etc/default/kbd 文件, 64-65

/etc/default/login 文件

安全 Shell, 294-295

登录缺省设置, 58

说明, 296

限制远程 root 访问, 62-63

/etc/default/su 文件

监视 su 命令, 61-62

/etc/default/su 文件 (续)

- 监视访问尝试, 62-63

- 显示 su 命令尝试, 62-63

- /etc/hosts.equiv 文件, 说明, 296

- /etc/krb5/kadm5.acl 文件, 说明, 461

- /etc/krb5/kadm5.keytab 文件, 说明, 461

- /etc/krb5/kdc.conf 文件, 说明, 461

- /etc/krb5/kpropd.acl 文件, 说明, 462

- /etc/krb5/krb5.conf 文件, 说明, 462

- /etc/krb5/krb5.keytab 文件, 说明, 462

- /etc/krb5/warn.conf 文件, 说明, 462

- /etc/logindefperm 文件, 40

- /etc/nologin 文件

- 临时禁止用户登录, 56-57

- 说明, 296

- /etc/pam.conf 文件, Kerberos 和, 462

- /etc/publickey 文件, DH 验证和, 248

- /etc/security/audit_event 文件, 审计事件和, 480

- /etc/security/device_allocate 文件, 88

- /etc/security/device_maps 文件, 87

- /etc/security/policy.conf 文件, 算法配置, 59-60

- /etc/ssh_host_dsa_key.pub 文件, 说明, 295

- /etc/ssh_host_key.pub 文件, 说明, 295

- /etc/ssh_host_rsa_key.pub 文件, 说明, 295

- /etc/ssh/shosts.equiv 文件, 说明, 296

- /etc/ssh/ssh_config 文件

- 覆盖, 296

- 关键字, 290-295

- 配置安全 Shell, 289

- 说明, 296

- 主机特定的参数, 294

- /etc/ssh/ssh_host_dsa_key 文件, 说明, 295

- /etc/ssh/ssh_host_key 文件, 覆盖, 296

- /etc/ssh/ssh_host_rsa_key 文件, 说明, 295

- /etc/ssh/ssh_known_hosts 文件

- 安全分发, 295

- 覆盖, 297

- 控制分发, 295

- 说明, 295

- /etc/ssh/sshd_config 文件

- 关键字, 290-295

- 说明, 295

- /etc/ssh/sshrdrc 文件, 说明, 296

/etc/syslog.conf 文件

- PAM 和, 259

- 可执行栈消息和, 116

- 审计和, 522, 555

- 失败的登录和, 57-59

- exec_args 审计标记

- argv 策略和, 564

- 格式, 564

- exec_attr 数据库

- 说明, 191

- 摘要, 189

- exec_env 审计标记, 格式, 564

- export 子命令, pktool 命令, 237-238

- EXTERNAL 安全机制插件, SASL 和, 266

F

- f 选项

- st_clean 脚本, 91

- 基于 Kerberos 的命令, 456, 457-458

- F 选项

- deallocate 命令, 87

- 基于 Kerberos 的命令, 456, 457-458

- FallBackToRsh 关键字, ssh_config 文件, 291

- fd_clean 脚本, 说明, 90

- fe 审计事件修饰符, 565

- file 审计标记, 格式, 564

- FILE 特权, 139

- find 命令, 查找文件setuid 权限, 123

- FIPS-140 支持, 使用 Sun Crypto Accelerator 6000 卡的
安全 Shell, 272

- flags 行, 进程预选掩码, 559

- fmri 审计标记, 格式, 565

- ForwardAgent 关键字, 安全 Shell 转发验证, 291

- ForwardX11 关键字, 安全 Shell 端口转发, 291

- ForwardX11Trusted 关键字, 安全 Shell 端口转
发, 291

- fp 审计事件修饰符, 565

- ftp 命令

- Kerberos 和, 455-457, 462

- 记录文件传输, 551-552

- 设置保护级别, 457

- ftpd 守护进程, Kerberos 和, 463

G

GatewayPorts 关键字, 安全 Shell, 291
 gencert 子命令, pktool 命令, 234–235
 getdevpolicy 命令, 说明, 85
 getent 命令, 说明, 192
 -getflags 选项
 auditconfig 命令, 503–504, 504–505
 -getnaflags 选项
 auditconfig 命令, 503–504, 504–505
 -getplugin 选项
 auditconfig 命令, 503–504, 520–521, 521–523
 -getpolicy 选项
 auditconfig 命令, 503–504, 508–511
 -getqctrl 选项, auditconfig 命令, 503–504
 gkadmin 命令
 另请参见 SEAM Tool
 说明, 463
 .gkadmin 文件
 SEAM Tool 和, 413
 说明, 461
 GlobalKnownHostsFile 关键字
 请参见 GlobalKnownHostsFile 关键字
 ssh_config 文件, 291
 group 审计标记
 格式, 565
 组策略和, 565
 group 审计策略
 group 标记和, 565
 和 groups 标记, 497
 说明, 497
 GSS-API
 Kerberos 和, 302
 安全 Shell 中的凭证, 288
 在安全 Shell 中验证, 270
 gssapi.so.1 插件, SASL 和, 266
 GSSAPIAuthentication 关键字, 安全 Shell, 291
 GSSAPIDelegateCredentials 关键字, ssh_config 文件, 291
 GSSAPIKeyExchange 关键字, 安全 Shell, 291
 GSSAPIStoreDelegatedCredentials 关键字,
 sshd_config 文件, 291
 gsscred 表, 使用, 473–474
 gsscred 命令, 说明, 463
 gssd 守护进程, Kerberos 和, 463–464

H

-h 选项, auditrecord 命令, 530
 HashKnownHosts 关键字, ssh_config 文件, 291
 header 审计标记
 格式, 565–566
 审计记录中的顺序, 565–566
 事件修饰符, 565
 hmac-md5 算法, ssh_config 文件, 292
 hmac-sha1 加密算法, ssh_config 文件, 292
 Host 关键字
 ssh_config 文件, 291, 294
 host 主体
 创建, 328, 334
 HostbasedAuthentication 关键字, 安全 Shell, 291
 HostbasedUsesNameFromPacketOnly 关键字,
 sshd_config 文件, 291
 HostKey 关键字, sshd_config 文件, 291
 HostKeyAlgorithms 关键字, ssh_config 文件, 291
 HostKeyAlias 关键字, ssh_config 文件, 291
 HostName 关键字, ssh_config 文件, 291
 hosts.equiv 文件, 说明, 296

I

-I 选项
 bart create 命令, 97
 st_clean 脚本, 90
 -i 选项
 bart create 命令, 97, 99
 encrypt 命令, 215
 st_clean 脚本, 90
 ID
 将 UNIX 映射到 Kerberos 主体, 473–474
 审计
 概述, 477–478
 机制, 559
 审计会话, 559
 IgnoreIfUnknown 关键字, ssh_config 文件, 291
 IgnoreRhosts 关键字, sshd_config 文件, 291
 IgnoreUserKnownHosts 关键字, sshd_config 文件, 292
 import 子命令, pktool 命令, 235–236
 in.ftpd 守护进程, Kerberos 和, 463
 in.rlogind 守护进程, Kerberos 和, 464

- in.rshd 守护进程, Kerberos 和, 464
- in.telnetd 守护进程, Kerberos 和, 464
- include 控制标志, PAM, 261
- install 子命令, cryptoadm 命令, 223
- INTERNAL 插件, SASL 和, 266
- Internet 防火墙设置, 49–50
- Internet 相关标记
 - ip address 标记, 566
 - ip port 标记, 566
 - socket 标记, 568–569
- ioctl() 系统调用, AUDIO_SETINFO(), 90
- ip address 审计标记, 格式, 566
- IP MIB-II, 从 /dev/arp 获取信息, 76
- ip port 审计标记, 格式, 566
- IP 地址
 - 安全 Shell 检查, 290
 - 安全 Shell 缺省值例外, 276–277
- IPC_perm 审计标记, 格式, 567
- ipc 类型字段值 (ipc 标记), 566–567
- ipc 审计标记, 566–567
 - 格式, 566–567
- IPC 特权, 139

K

- k 选项
 - encrypt 命令, 215
 - mac 命令, 213
 - 基于 Kerberos 的命令, 456
- K 选项
 - encrypt 命令, 215
 - mac 命令, 214
 - rolemod 命令, 168
 - usermod 命令, 169
 - 基于 Kerberos 的命令, 456
- .k5.REALM 文件, 说明, 462
- .k5login 文件
 - 而不显示口令, 454
 - 说明, 453–455, 461
- kadm5.acl 文件
 - 说明, 461
 - 条目的格式, 428
 - 新主体和, 422, 424
 - 主 KDC 项, 327, 333, 371
- kadm5.keytab 文件, 说明, 461
- kadmin.local 命令
 - 说明, 463
 - 添加管理主体, 327, 333
 - 自动创建主体, 416
- kadmin.log 文件, 说明, 462
- kadmin 命令
 - ktadd 命令, 442–443
 - ktremove 命令, 443
 - SEAM Tool 和, 411
 - 创建 host 主体, 328, 334
 - 从密钥表删除主体, 443–444
 - 说明, 463
- kadmind 守护进程
 - Kerberos 和, 463
 - 主 KDC 和, 464
- kbd 文件, 64–65
- KbdInteractiveAuthentication 关键字, 安全 Shell, 292
- kcfd 守护进程, 203, 230
- kcliclient 命令, 说明, 463
- kdb5_ldap_util 命令, 说明, 463
- kdb5_util 命令
 - 创建 KDC 数据库, 327
 - 创建存储文件, 340, 381
 - 说明, 463
- KDC
 - 备份和传播, 372–373
 - 创建 host 主体, 328, 334
 - 创建数据库, 327
 - 从, 316
 - 定义, 464
 - 从或主, 307, 322
 - 端口, 316
 - 规划, 316
 - 将管理文件从从 KDC 服务器复制到主 KDC 服务器, 338, 380
 - 交换主与从, 368–372
 - 配置从
 - 交互, 336
 - 手动, 337–340
 - 自动, 335–336
 - 配置主
 - 交互, 324–325

KDC, 配置主 (续)

- 使用 LDAP, 329–335

- 手动, 325–329

- 自动, 323–324

- 启动守护进程, 340, 381

- 数据库传播, 317

- 同步时钟

- 从 KDC, 340, 381

- 主 KDC, 329, 335

- 限制对服务器的访问, 389–390

- 主

- 定义, 464

kdc.conf 文件

- 票证生命周期和, 466

- 说明, 461

kdc.log 文件, 说明, 462

kdcmgr 命令

- 服务器状态, 325

- 配置从

- 交互, 336

- 自动, 335

- 配置主

- 交互, 324

- 自动, 323

kdestroy 命令

- Kerberos 和, 462

- 示例, 450

KeepAlive 关键字, 安全 Shell, 292

Kerberos

- Kerberos V5 协议, 301

- 参考, 461–474

- 错误消息, 393–406

- 概述

- 基于 Kerberos 的命令, 455–457

- 验证系统, 302–307, 468

- 故障排除, 406

- 管理, 411–446

- 管理工具

- 请参见 SEAM Tool

- 规划, 313–320

- 获取对服务器的访问权限, 469–471

- 基于 Kerberos 的命令的选项, 456

- 加密类型

- 概述, 320

Kerberos, 加密类型 (续)

- 使用, 472–473

- 仅启用基于 Kerberos 的应用程序, 389

- 口令管理, 450–455

- 联机帮助, 320

- 领域

- 请参见领域 (Kerberos)

- 命令, 455–460, 462–463

- 配置 KDC 服务器, 322–340

- 配置决策, 313–320

- 使用, 447–460

- 使用基于 Kerberos 的命令的示例, 459–460

- 守护进程, 463–464

- 网络命令选项表, 457

- 文件, 461–462

- 向您的帐户授予访问权限, 453–455

- 远程应用程序, 305

- 术语, 464–468

- 组件, 309–310

Kerberos 命令, 455–460

- 仅启用基于 Kerberos 的, 389

- 示例, 459–460

Kerberos 验证, 和安全 RPC, 248

- kern.notice 项, syslog.conf 文件, 116

- KEYBOARD_ABORT 系统变量, 64–65

- keylogin 命令, 用于安全 RPC, 249

- KeyRegenerationInterval 关键字, sshd_config 文件, 292

- keyserv 守护进程, 251–252

- keyserver

- 启动, 251–252

- 说明, 249

- keytab 选项, SASL 和, 267

- kgcmgr 命令, 说明, 463

- kinit 命令

- F 选项, 448

- Kerberos 和, 462

- 票证生命周期, 466

- 示例, 448

- klist 命令

- f 选项, 448–450

- Kerberos 和, 462

- 示例, 448–450

KMF

创建

- 密钥库的口令短语, 233

- 密钥库口令, 238

- 自签名证书, 234–235

- 导出证书, 237–238

管理

- PKI 策略, 232

- 插件, 232

- 公钥技术 (public key technology, PKI), 231

- 密钥库, 233

- 将证书导入密钥库, 235–236

- 库, 231

- 列出插件, 243–244

- 密钥库, 231, 233

- 删除插件, 243–244

- 实用程序, 232

- 添加插件, 243–244

kmfcfg 命令

- list plugin 子命令, 243–244

- 插件子命令, 231, 232

known_hosts 文件

- 控制分发, 295

- 说明, 296

- Korn shell, 特权版本, 136

kpasswd 命令

- Kerberos 和, 463

- passwd 命令和, 451

- 错误消息, 452

- 示例, 452

- kprop 命令, 说明, 463

- kpropd.acl 文件, 说明, 462

- kpropd 守护进程, Kerberos 和, 463

- kproplog 命令, 说明, 463

krb5.conf 文件

- domain_realm 部分, 315

- 编辑, 325, 331

- 端口定义, 316

- 说明, 462

- krb5.keytab 文件, 说明, 462

- krb5cc_uid 文件, 说明, 462

krb5kdc 守护进程

- Kerberos 和, 464

- 启动, 340, 381

krb5kdc 守护进程 (续)

- 主 KDC 和, 464

- ksh 命令, 特权版本, 136

ktadd 命令

- 添加服务主体, 441, 442–443

- 语法, 442

- ktkt_warnd 守护进程, Kerberos 和, 464

- ktremove 命令, 443

ktutil 命令

- delete_entry 命令, 445

- Kerberos 和, 463

- list 命令, 444, 445

- read_kt 命令, 444, 445

- 查看主体列表, 443, 444–445

- 管理密钥表文件, 441

- kvno 命令, Kerberos 和, 463

L

- L 选项, ssh 命令, 282

-l 选项

- digest 命令, 212

- encrypt 命令, 206

- mac 命令, 213

- LDAP, 配置主 KDC 服务器, 329–335

LDAP 命名服务

- 口令, 37

- 指定口令算法, 60–61

list_devices 命令

- 所需授权, 87

- 要求授权, 193

- list plugin 子命令, kmfcfg 命令, 243–244

- list 命令, 444, 445

- list 子命名, pktool 命令, 234

- ListenAddress 关键字, sshd_config 文件, 292

- LocalForward 关键字, ssh_config 文件, 292

- log_level 选项, SASL and, 267

- logadm 命令, 归档文本摘要审计文件, 538

- login 环境变量, 安全 Shell, 294–295

login 文件

- 登录缺省设置, 58

- 限制远程 root 访问, 62–63

- LoginGraceTime 关键字, sshd_config 文件, 292

- loginlog 文件, 保存失败的登录尝试, 57

logins 命令

显示没有口令的用户, 56

显示用户的登录状态, 55

语法, 55

LogLevel 关键字, 安全 Shell, 292

LookupClientHostnames 关键字, sshd_config 文件, 292

-lspolicy 选项, auditconfig 命令, 508–511

M

-M 选项, auditreduce 命令, 533

-m 选项

cryptoadm 命令, 223, 225

基于 Kerberos 的命令, 456

mac 命令

说明, 203

语法, 213

MACS 关键字, 安全 Shell, 292

Match 关键字, sshd_config 文件, 292

Match 块, 安全 Shell 缺省值例外, 276–277

max_life 值, 说明, 466

max_renewable_life 值, 说明, 467

MaxStartups 关键字, sshd_config 文件, 292

MD4 加密算法, 内核提供者, 219

MD5 加密算法

policy.conf 文件, 59–60, 60

MD5 加密算法, 内核提供者, 219

MD5 加密算法

允许在异构环境中, 60

mech_dh 机制, GSS-API 凭证, 288

mech_krb 机制, GSS-API 凭证, 288

mech_list 选项, SASL 和, 266

Media Backup (介质备份) 权限配置文件

指定给可信用户, 129, 155

Media Restore (介质恢复) 权限配置文件, 指定给可信用户, 155

messages 文件, 可执行栈消息, 116

metaslot

管理, 203

加密框架中的定义, 202

mount 命令, 带安全属性, 78

mt 命令, 磁带设备清除和, 90

N

-n 选项, bart create 命令, 97

n2cp 驱动程序

加密框架的硬件插件, 201

列出机制, 227–228

na 审计事件修饰符, 565

ncp 驱动程序

加密框架的硬件插件, 201

列出机制, 227–228

NET 特权, 139

netservices limited 安装选项, 44

never-audit 类, 进程预选掩码, 559

newkey 命令

生成密钥, 249

为 NIS 用户创建密钥, 253–254

NFS 服务器, 为 Kerberos 配置, 346–348

NFS 文件系统

提供客户机-服务器安全性, 248–251

通过 AUTH_DH 进行安全访问, 254

验证, 247

NIS 命名服务

口令, 37

验证, 247

指定口令算法, 60

nisaddcred 命令, 生成密钥, 249

nobody 用户, 47

noexec_user_stack_log 变量, 116, 124

noexec_user_stack 变量, 116, 124

NoHostAuthenticationForLocalHost 关键字, ssh_config 文件, 292

nologin 文件, 说明, 296

nscd (名称服务高速缓存守护进程), 使用, 192

NSS, 管理密钥库, 233

NTP

Kerberos 规划与, 318

从 KDC 服务器和, 340, 381

主 KDC 和, 329, 335

NumberOfPasswordPrompts 关键字, ssh_config 文件, 292

O

-o 选项

auditreduce 命令, 532–533, 533, 534

-o 选项, encrypt 命令, 215

OpenSSH, 请参见安全 Shell

OpenSSL

版本, 232

管理密钥库, 233

Operator (操作员) (RBAC)

建议的角色, 129

权限配置文件, 186

optional 控制标志, PAM, 261

ovsec_admin.xxxxx 文件, 说明, 462

P

-p 选项

auditrecord 命令, 531

bart create, 99

cryptoadm 命令, 224, 225

logins 命令, 56

PAM

/etc/syslog.conf 文件, 259

Kerberos 和, 310

概述, 255

规划, 257

框架, 256

配置文件

Kerberos 和, 462

堆栈示例, 263

堆栈说明, 260

堆栈图, 261

介绍, 259

控制标志, 261

语法, 259

任务列表, 257

添加模块, 258

用于高速缓存验证的堆栈, 152

pam.conf 文件, 请参见 PAM 配置文件

pam_roles 命令, 说明, 192

pam_tty_tickets.so.1 模块, PAM, 152

PAM 模块, 152

PAM 配置文件, 添加 su 堆栈, 152

PAMServiceName 关键字, sshd_config 文件, 292

PAMServicePrefix 关键字, sshd_config 文件, 292

passwd 命令

更改角色的口令, 166–167

passwd 命令 (续)

和 kpasswd 命令, 451

和命名服务, 37

语法, 54

PasswordAuthentication 关键字, 安全 Shell, 292

path_attr 审计标记, 567

PATH 环境变量

和安全性, 43

设置, 43

path 审计标记, 格式, 567

path 审计策略, 说明, 497

PermitEmptyPasswords 关键字, sshd_config 文件, 292

PermitRootLogin 关键字, sshd_config 文件, 292

PermitUserEnvironment 关键字, sshd_config 文件, 292

perzone 审计策略

何时使用, 487–488

设置, 510–511

使用, 493, 525–526, 556

说明, 497

pfcsch 命令, 说明, 136

pfexec 命令, 说明, 192

pfksh 命令, 说明, 136

pfsh 命令, 说明, 136

PidFile 关键字, 安全 Shell, 292

PKCS #10 CSR

签署

使用 pktool 命令, 242–243

PKCS #11 库

添加提供者库, 223

在加密框架中, 201

PKCS #11 软令牌, 管理密钥库, 233

PKCS #12 文件, 保护, 237

pkcs11_kernel.so 用户级提供者, 219

pkcs11_softtoken.so 用户级提供者, 219

PKI

由 KMF 管理, 231

由 KMF 管理策略, 232

pktool 命令

export 子命令, 237–238

gencert 子命令, 234–235

import 子命令, 235–236

list 子命名, 234

pktool 命令 (续)

- setpin 子命令, 238
- 创建自签名证书, 234-235
- 管理 PKI 对象, 231
- 签署 PKCS #10 CSR, 242-243
- 生成密钥, 208-212
- 生成密钥对, 238-242

plain.so.1 插件, SASL 和, 266

plugin_list 选项, SASL 和, 267

policy.conf 文件

关键字

- 用于口令算法, 38
- 用于特权, 194
- 针对 RBAC 授权, 191
- 针对工作站所有者, 191
- 针对权限配置文件, 191
- 针对特权, 191

说明, 191, 192

指定加密算法, 59-60

指定口令算法, 59-60

在命名服务中, 60

Port 关键字, 安全 Shell, 292

ppriv 命令

列出特权, 180

用于调试, 181

praudit 命令

XML 格式, 536

查看审计记录, 535-537

将审计记录转换为可读格式, 536

说明, 555

在脚本中使用, 536

传输 auditreduce 输出到, 536

PreferredAuthentications 关键字, ssh_config 文件, 292

PreUserauthHook 关键字, ssh_config 文件, 292

principal.kadm5.lock 文件, 说明, 462

principal.kadm5 文件, 说明, 462

principal.ok 文件, 说明, 462

principal.ulong 文件, 说明, 462

principal 文件, 说明, 462

Printer Management (打印机管理) (RBAC), 权限配置文件, 186

PrintLastLog 关键字, ssh_config 文件, 293

PrintMotd 关键字, sshd_config 文件, 293

priv.debug 项, syslog.conf 文件, 195

PRIV_DEFAULT 关键字

policy.conf 文件, 191, 195

PRIV_LIMIT 关键字

policy.conf 文件, 191, 195

PRIV_PROC_LOCK_MEMORY 特权, 140

private 保护级别, 457

privilege 审计标记, 567

PROC 特权, 139

process 审计标记, 格式, 568

prof_attr 数据库

summary, 189

说明, 190-191

profiles 命令, 说明, 192

PROFS_GRANTED 关键字, policy.conf 文件, 191

project.max-locked-memory 资源控制, 140

PROM 安全模式, 63-65

Protocol 关键字, 安全 Shell, 293

ProxyCommand 关键字, ssh_config 文件, 293

pseudo-tty, 在安全 Shell 中使用, 289

PubkeyAuthentication 关键字, 安全 Shell, 293

public 审计策略

说明, 497

只读事件, 497

publickey 映射, DH 验证, 248-251

pwcheck_method 选项, SASL 和, 266

Q

qsize 属性, 审计插件, 511-512

R

-R 选项

bart create, 97, 99

ssh 命令, 282

-r 选项

bart create, 99

passwd 命令, 37

RBAC

保护脚本, 162

查看权限, 147-149

查看所有 RBAC 安全属性, 146-147

RBAC (续)

- 创建权限配置文件, 159–161
 - 更改角色口令, 166–167
 - 故障排除, 162–165
 - 管理命令, 192
 - 规划, 153–155
 - 获取管理权限, 150–152
 - 基本概念, 130–132
 - 检查脚本或程序的授权, 162
 - 将用户限于桌面应用程序, 170–171
 - 命名服务和, 189
 - 配置, 153–165
 - 配置文件 shell, 136
 - 权限配置文件, 135
 - 权限配置文件数据库, 190–191
 - 缺省值, 146–152
 - 审计角色, 159
 - 审计配置文件, 556
 - 使用用户口令承担角色, 172–173
 - 使用用户口令来使用权限配置文件, 173
 - 授权, 133–134
 - 授权数据库, 190
 - 数据库, 189–191
 - 添加角色, 155–157
 - 添加特权用户, 169
 - 限制权限, 171–172
 - 修改角色, 167–168
 - 修改用户, 168–169
 - 用于管理的命令, 192
 - 与超级用户模型比较, 127–130
 - 元素, 130–132
- RC4, 请参见ARCFOUR 内核提供者
- rcp 命令
- Kerberos 和, 455–457, 463
- rd 审计事件修饰符, 565
- read_kt 命令, 444, 445
- reauth_timeout 选项, SASL 和, 267
- RekeyLimit 关键字, ssh_config 文件, 293
- rem_drv 命令, 说明, 85
- RemoteForward 关键字, ssh_config 文件, 293
- required 控制标志, PAM, 261
- requisite 控制标志, PAM, 261
- RETRIES 安全 Shell 中的, 294
- return 审计标记, 格式, 568
- rewoffl 选项
- mt 命令
 - 磁带设备清除和, 90
- .rhosts 文件, 说明, 296
- RhostsAuthentication 关键字, 安全 Shell, 293
- RhostsRSAAuthentication 关键字, 安全 Shell, 293
- rlogin 命令
- Kerberos 和, 455–457, 463
- rlogind 守护进程, Kerberos 和, 464
- roleadd 命令
- 使用, 155
 - 说明, 192
- roleauth 关键字, 角色的口令, 172–173
- rolemod 命令
- 更改角色的属性, 167, 172
 - 角色的口令, 172–173
 - 说明, 192
- roles 命令
- 使用, 149
 - 说明, 192
- root 角色
- 从 root 用户进行更改, 174
- root 角色
- 更改口令, 54
- root 角色
- 更改为 root 用户, 173–175
- root 角色
- 提供的角色, 129
- root 角色 (RBAC)
- 承担角色, 150
 - 故障排除, 175
- root 用户
- 跟踪登录, 42
 - 更改为 root 角色, 174
 - 监视 su 命令尝试, 61–62
 - 监视 su 命令意图, 42
 - 限制访问, 47
 - 限制远程访问, 62–63
 - 在 RBAC 中替换, 136
 - 在控制台中显示访问尝试, 62–63
- root 帐户, 说明, 39
- root 主体, 添加至主机的密钥表, 441
- RSA 内核提供者, 219
- RSAAuthentication 关键字, 安全 Shell, 293

rsh 命令

- Kerberos 和, 455–457, 463
- rsh 命令 (受限 shell), 43
- rshd 守护进程, Kerberos 和, 464
- rstchown 系统变量, 119
- rules 文件 (BART), 95
- rules 文件格式 (BART), 106–107
- 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, st_clean 脚本, 91
- s 选项
 - audit 命令, 526–528, 529
- safe 保护级别, 457
- SASL
 - 插件, 266
 - 概述, 265
 - 环境变量, 266
 - 选项, 266–267
- saslauthd_path 选项, SASL 和, 267
- scp 命令
 - 复制文件, 283
 - 说明, 297
- SCSI 设备, st_clean 脚本, 89
- SEAM Tool
 - gkadmin 命令, 411
 - .gkadmin 文件, 413
 - kadmin 命令, 411
 - 帮助, 413
 - 帮助内容, 413
 - 查看策略列表, 429–431
 - 查看策略属性, 431–433
 - 查看主体列表, 417–419
 - 查看主体属性, 419–421
 - 创建新策略, 421, 433–434
 - 创建新主体, 421–423
 - 登录窗口, 415
 - 等效命令行, 412–413
 - 复制主体, 424
 - 概述, 412–415
 - 过滤模式字段, 418
 - 和 X 窗口系统, 412–413

SEAM Tool (续)

- 和列表特权, 440
- 和受限管理特权, 440–441
- 或 kadmin 命令, 412
- 联机帮助, 413
- 面板表, 437–440
- 面板说明, 437–440
- 启动, 415
- 缺省值, 415
- 删除策略, 436
- 删除主体, 425–426
- 上下文有关帮助, 413
- 设置主体缺省值, 426–427
- 受特权的影响, 440
- 特权, 440
- 显示主体子列表, 418
- 修改策略, 435–436
- 修改的文件, 413
- 修改主体, 424–425
- SEAM Tool 的等效命令行, 412–413
- sendmail 命令, 要求授权, 193
- seq 审计策略
 - 和 sequence 标记, 497, 568
 - 说明, 497
- sequence 审计标记
 - 格式, 568
 - 和 seq 审计策略, 568
- ServerAliveCountMax 关键字, ssh_config 文件, 293
- ServerAliveInterval 关键字, ssh_config 文件, 293
- ServerKeyBits 关键字, sshd_config 文件, 293
- setflags 选项, auditconfig 命令, 504–505
- setgid 权限
 - 安全风险, 112
 - 符号模式, 114
 - 绝对模式, 114, 122
 - 说明, 112
- setnaflags 选项, auditconfig 命令, 504–505
- setpin 子命令, pktool 命令, 238
- setplugin 选项
 - auditconfig 命令, 520–521, 521–523
- setpolicy 选项, auditconfig 命令, 508–511
- setuid 权限
 - 安全风险, 44, 112
 - 查找设置了权限的文件, 123

setuid 权限 (续)

- 符号模式, 114
- 绝对模式, 114, 122
- 说明, 111-112

sftp 命令

- 复制文件, 283
- 审计文件传输, 551-552
- 说明, 297

sh 命令, 特权版本, 136

SHA1 内核提供者, 219

SHA2 内核提供者, 219

shell, 特权版本, 136

shell 脚本, 写入特权, 183

shell 进程, 列出其特权, 180-181

shell 命令, 传递父 shell 进程号, 180

shosts.equiv 文件, 说明, 296

.shosts 文件, 说明, 296

slave_datatrans_slave 文件, 说明, 462

slave_datatrans 文件

- KDC 传播和, 372-373
- 说明, 462

SMF

- auditd 服务, 553-554
- kcfd 服务, 203
- ssh 服务, 276
- 管理缺省安全配置, 44
- 加密框架服务, 203
- 启用 keyserver, 251
- 设备分配服务, 86
- 重新启动加密框架, 230

socket 审计标记, 568-569

solaris.device.revoke 授权, 87

sp 审计事件修饰符, 565

sr_clean 脚本, 说明, 90

ssh-add 命令

- 存储私钥, 280-281
- 示例, 280-281, 281
- 说明, 297

ssh-agent 命令

- 从命令行, 280-281
- 说明, 297

.ssh/config 文件

- 覆盖, 296
- 说明, 296

ssh_config 文件

- 覆盖, 296
- 关键字, 290-295
- 请参见特定关键字
- 配置安全 Shell, 289
- 主机特定的参数, 294

.ssh/environment 文件, 说明, 296

ssh_host_dsa_key.pub 文件, 说明, 295

ssh_host_dsa_key 文件, 说明, 295

ssh_host_key.pub 文件, 说明, 295

ssh_host_key 文件, 覆盖, 296

ssh_host_rsa_key.pub 文件, 说明, 295

ssh_host_rsa_key 文件, 说明, 295

.ssh/id_dsa 文件, 297

.ssh/id_rsa 文件, 297

.ssh/identity 文件, 297

ssh-keygen 命令

- 口令短语保护, 271
- 使用, 277-279
- 说明, 297

ssh-keyscan 命令, 说明, 297

ssh-keysign 命令, 说明, 297

.ssh/known_hosts 文件

- 覆盖, 297
- 说明, 296

ssh_known_hosts 文件, 295

.ssh/rc 文件, 说明, 296

ssh 命令

- 端口转发选项, 282
- 覆盖关键字设置, 297
- 使用, 279-280
- 使用代理命令, 285
- 说明, 297

sshd_config 文件

- /etc/default/login 项的覆盖, 294-295
- 关键字, 290-295
- 请参见特定关键字
- 说明, 295

sshd.pid 文件, 说明, 295

sshd 命令, 说明, 297

sshrd 文件, 说明, 296

st_clean 脚本

- 说明, 90
- 用于磁带机, 89

- sticky 位权限
 - 符号模式, 114
 - 绝对模式, 114, 122
 - 说明, 112
 - Stop (停止) (RBAC), 权限配置文件, 186
 - StrictHostKeyChecking 关键字, ssh_config 文件, 293
 - StrictModes 关键字, sshd_config 文件, 293
 - su 命令
 - 监视使用, 61–62
 - 在角色承担中, 149–150
 - 在控制台中显示访问尝试, 62–63
 - su 文件, 监视 su 命令, 61–62
 - subject 审计标记, 格式, 569
 - Subsystem 关键字, sshd_config 文件, 293
 - sufficient 控制标志, PAM, 261
 - suolog 文件, 61–62
 - 监视内容, 61
 - Sun Crypto Accelerator 1000 板, 列出机制, 228–230
 - Sun Crypto Accelerator 6000 板
 - 加密框架的硬件插件, 201
 - 列出机制, 227–228
 - Sun Crypto Accelerator 6000 卡, 安全 Shell 和 FIPS-140, 272
 - svc:/system/device/allocate, 设备分配服务, 86
 - svcadm 命令
 - 管理加密框架, 202, 203
 - 启用 keyserver 守护进程, 252
 - 启用加密框架, 230
 - 刷新加密框架, 222–223
 - 重新启动
 - syslog 守护进程, 58, 522
 - 安全 Shell, 276
 - svcs 命令
 - 列出 keyserver 服务, 251
 - 列出加密服务, 230
 - SYS 特权, 139
 - syslog.conf 文件
 - audit.notice 级别, 522
 - kern.notice 级别, 116
 - priv.debug 项, 195
 - 保存失败的登录尝试, 57–59
 - 和审计, 555
 - 可执行栈消息, 116
 - syslog.conf 文件 (续)
 - 特权调试, 194
 - SYSLOG_FAILED_LOGINS
 - 安全 Shell 中的, 294
 - 系统变量, 58
 - SyslogFacility 关键字, sshd_config 文件, 293
 - System Administrator (系统管理员) (RBAC)
 - 保护硬件, 64
 - 建议的角色, 129
 - 权限配置文件, 186
 - System V IPC
 - IPC_perm 审计标记, 567
 - ipc 审计标记, 566–567
 - /system/volatile/sshd.pid 文件, 说明, 295
- ## T
- T 选项
 - encrypt 命令, 216
 - mac 命令, 214
 - t 选项, audit 命令, 528–529
 - tail 命令, 使用示例, 500
 - TCP
 - 安全 Shell 和, 275, 289
 - 地址, 566
 - telnet 命令
 - Kerberos 和, 455–457, 463
 - telnetd 守护进程, Kerberos 和, 464
 - text 审计标记, 格式, 569
 - TGS, 获取凭证, 469
 - TGT, 在 Kerberos 中, 303–304
 - /tmp/krb5cc_uid 文件, 说明, 462
 - /tmp/ovsec_admin.xxxxx 文件, 说明, 462
 - TMPFS 文件系统, 安全, 112
 - trail 审计策略
 - 和 trailer 标记, 498
 - 说明, 498
 - trailer 审计标记
 - praudit 显示, 569
 - 格式, 569
 - 审计记录中的顺序, 569
 - truss 命令, 用于特权调试, 181

U

-U 选项, allocate 命令, 87

UDP

安全 Shell, 275

地址, 566

端口转发, 275

用于远程审计日志, 483

umask 值

典型值, 113

和文件创建, 112-113

umount 命令, 带安全属性, 78

UNIX 文件权限, 请参见文件, 权限

update_drv 命令

使用, 75

说明, 85

use_authid 选项, SASL 和, 267

use of authorization 审计标记, 570

use of privilege 审计标记, 570

UseOpenSSLEngine 关键字, 安全 Shell, 293

UsePrivilegedPort 关键字, 安全 Shell, 293

user_attr 数据库

列出审计预选的用户例外, 505-508

说明, 189

user_attr 文件, 系统范围的审计类的例外, 481

User Security (用户安全) 权限配置文件, 为用户修改审计预选, 505-508

User 关键字, ssh_config 文件, 293

user 审计标记, 570

useradd 命令, 说明, 192

userattr 命令

说明, 192

显示系统范围的审计的例外, 503-504

userdel 命令, 说明, 192

UserKnownHostsFile 关键字, ssh_config 文件, 293

UserKnownHostsFile2 关键字, 请参

见 UserKnownHostsFile 关键字

usermod 命令

audit_flags 关键字, 505-508

usermod 命令

更改用户的 RBAC 属性, 168

仅将用户限于桌面图标, 171

说明, 192

usermod 命令

为 audit_flags 例外使用插入记号 (^) 前缀, 506-507

usermod 命令

系统范围的审计的例外, 481

用于指定角色, 157-158

usermod 命令

指定审计预选的用户例外, 505-508

UseRsh 关键字, ssh_config 文件, 293

/usr/bin/ftp 命令, Kerberos 和, 462

/usr/bin/kdestroy 命令, Kerberos 和, 462

/usr/bin/kinit 命令, Kerberos 和, 462

/usr/bin/klist 命令, Kerberos 和, 462

/usr/bin/kpasswd 命令, Kerberos 和, 463

/usr/bin/ktutil 命令, Kerberos 和, 463

/usr/bin/kvno 命令, Kerberos 和, 463

/usr/bin/rcp 命令, Kerberos 和, 463

/usr/bin/rlogin 命令, Kerberos 和, 463

/usr/bin/rsh 命令, Kerberos 和, 463

/usr/bin/telnet 命令, Kerberos 和, 463

/usr/lib/kprop 命令, 说明, 463

/usr/lib/krb5/kadmind 守护进程, Kerberos 和, 463

/usr/lib/krb5/kpropd 守护进程, Kerberos 和, 463

/usr/lib/krb5/krb5kdc 守护进程, Kerberos 和, 464

/usr/lib/krb5/ktkt_warnd 守护进程, Kerberos 和, 464

/usr/lib/libsas1.so 库, 概述, 265

/usr/sbin/gkadmin 命令, 说明, 463

/usr/sbin/gsscred 命令, 说明, 463

/usr/sbin/in.ftpd 守护进程, Kerberos 和, 463

/usr/sbin/in.rlogind 守护进程, Kerberos 和, 464

/usr/sbin/in.rshd 守护进程, Kerberos 和, 464

/usr/sbin/in.telnetd 守护进程, Kerberos 和, 464

/usr/sbin/kadmin.local 命令, 说明, 463

/usr/sbin/kadmin 命令, 说明, 463

/usr/sbin/kclient 命令, 说明, 463

/usr/sbin/kdb5_ldap_util 命令, 说明, 463

/usr/sbin/kdb5_util 命令, 说明, 463

/usr/sbin/kgcmgr 命令, 说明, 463

/usr/sbin/kproplog 命令, 说明, 463

V

v1 协议, 安全 Shell, 269

v2 协议, 安全 Shell, 269

-v 选项

digest 命令, 212

mac 命令, 213

ppriv 命令, 180

/var/adm/auditlog 文件, 文本审计记录, 522

/var/adm/loginlog 文件, 保存失败的登录尝试, 57

/var/adm/messages 文件

可执行栈消息, 116

审计的故障排除, 541

/var/adm/sulog 文件, 监视内容, 61

/var/krb5/.k5.REALM 文件, 说明, 462

/var/krb5/kadmin.log 文件, 说明, 462

/var/krb5/kdc.log 文件, 说明, 462

/var/krb5/principal.kadm5.lock 文件, 说明, 462

/var/krb5/principal.kadm5 文件, 说明, 462

/var/krb5/principal.ok 文件, 说明, 462

/var/krb5/principal.ulog 文件, 说明, 462

/var/krb5/principal 文件, 说明, 462

/var/krb5/slave_datatrans_slave 文件, 说明, 462

/var/krb5/slave_datatrans 文件, 说明, 462

/var/log/authlog 文件, 失败的登录, 57-59

/var/log/syslog 文件, 审计的故障排除, 541

VerifyReverseMapping 关键字, ssh_config 文件, 293

vnode 审计标记, 格式, 563

W

warn.conf 文件, 说明, 462

wr 审计事件修饰符, 565

X

X.509 v3 证书, 生成, 242-243

-X 选项, 基于 Kerberos 的命令, 457

X 窗口系统, 和 SEAM Tool, 412-413

X11 转发

安全 Shell 中, 289

在 ssh_config 文件中配置, 291

X11DisplayOffset 关键字, sshd_config 文件, 293

X11Forwarding 关键字, sshd_config 文件, 293

X11UseLocalHost 关键字, sshd_config 文件, 294

-x 选项, 基于 Kerberos 的命令, 456

xauth 命令, X11 转发, 294

XAuthLocation 关键字, 安全 Shell 端口转发, 294

xclient 审计标记, 570

XML 格式, 审计记录, 536

Xylogics 磁带机设备清除脚本, 89

Z

ZFS File System Management (ZFS 文件系统管

理) 权限配置文件, 创建审计文件系统, 515-518

ZFS Storage Management (ZFS 存储管理) 权限配置

文件, 为审计文件创建池, 515-518

ZFS 文件系统, 为二进制审计文件创建, 515-518

zone.max-locked-memory 资源控制, 140

zonename 审计标记, 570

zonename 审计策略

使用, 493, 556

说明, 498

安

安全

保护 PROM, 63-65

保护硬件, 63-65

策略概述, 32

防止远程登录, 62-63

审计和, 485

系统硬件, 63-65

安全 NFS, 248

安全 RPC

keyserver, 249

概述, 48-49

和 Kerberos, 248

实现, 248-251

说明, 247

替代项, 49

安全 Shell

scp 命令, 283

TCP, 275

本地端口转发, 282

创建密钥, 277-279

当前发行版中的变更, 271-272

安全 Shell (续)

- 登录环境变量, 294–295
- 登录远程主机, 279–280
- 典型会话, 287–289
- 复制文件, 283
- 更改口令短语, 279
- 公钥验证, 270
- 关键字, 290–295
- 管理, 287–289
- 管理员任务列表, 272–273, 273
- 基础, OpenSSH, 271–272
- 跨防火墙连接, 283
- 连接防火墙外部
 - 从命令行, 285
 - 从配置文件, 283–285
- 命令执行, 289
- 命名标识文件, 295
- 配置端口转发, 275–276
- 配置服务器, 289
- 配置客户机, 289
- 生成密钥, 277–279
- 使用端口转发, 282
- 数据转发, 289
- 说明, 269
- 文件, 295
- 协议版本, 269
- 验证
 - 要求, 270–271
- 验证步骤, 288
- 验证方法, 270–271
- 以较少的提示登录, 280–281
- 用户过程, 277
- 远程端口转发, 282
- 在没有口令时使用, 280–281
- 指定系统缺省值例外, 276–277
- 转发邮件, 282

安全 Shell 中的 ALTSHELL, 294

安全 Shell 中的 CONSOLE, 294

安全 Shell 中的 PASSREQ, 294

安全 Shell 中的 PATH, 294

安全 Shell 中的 SUPATH, 295

安全 Shell 中的 TIMEOUT, 294

安全 Shell 中的 TZ, 294

安全策略, 缺省 (RBAC), 189

安全服务, Kerberos 和, 308

安全机制, 使用 -m 选项指定, 456

安全连接, 跨防火墙, 283

安全模式, 设置环境使用多个, 350–351

安全属性

- Network Security (网络安全) 权限配置文
件, 132
- 检查, 134
- 列出所有 RBAC, 146–147
- 命令的特殊 ID, 134
- 命令特权, 134
- 说明, 130
- 搜索顺序, 187
- 用于挂载分配的设备, 78
- 直接指定时的可用性注意事项, 137
- 直接指定时的注意事项, 137

安全性

- BART, 93–108
- DH 验证, 248–251
- netservices limited 安装选项, 44
- NFS 客户机-服务器, 248–251
- 安全 Shell, 269–285
- 安装选项, 44
- 保护设备, 89–91
- 防御拒绝服务攻击, 44
- 防御特洛伊木马, 43
- 计算文件的 MAC, 213–215
- 计算文件摘要, 212–213
- 加密框架, 199–204
- 加密文件, 215–218
- 口令加密, 37
- 跨不安全的网络, 283
- 密钥管理框架, 231–244
- 缺省安全, 44
- 设备, 40–42
- 设备分配, 73–91
- 审计, 477–489
- 系统, 35

安装, 缺省安全, 44

帮

帮助

SEAM Tool, 413

帮助 (续)

联机 URL, 320
帮助内容, SEAM Tool, 413

包

包传送
包粉碎, 50
防火墙安全性, 49

保

保存, 失败的登录尝试, 57
保护
 BIOS, 指针, 64
 PROM, 64
 登录任务列表, 54
 脚本, 162
 口令任务列表, 54
 密钥库内容, 237
 使用加密框架保护文件, 205–206
 通过使用带加密框架的口令, 233–234
 网络, 在安装时, 44
 系统, 防止危险程序, 122–123
保护级别
 clear, 457
 private, 457
 safe, 457
 在 ftp 中设置, 457
保护连接, 登录, 279–280
保护文件
 使用 UFS ACL, 115–116
 使用 UNIX 权限, 109–115, 117
 使用 UNIX 权限任务列表, 117
 用户过程, 117
保密性
 Kerberos 和, 301
 安全服务, 308
 可用性, 457

报

报告, BART, 93
报告工具, 请参见 bart compare

备

备份
 Kerberos 数据库, 372–373
 从 KDC, 316

变

变量
 KEYBOARD_ABORT, 64–65
 login 和安全 Shell, 294–295
 noexec_user_stack, 116
 noexec_user_stack_log, 116
 rstchown, 119
 安全 Shell 中的设置, 294
 代理服务器和端口, 284
 审计与命令关联的, 563–564
 添加到审计记录, 496, 564

标

标识文件 (安全 Shell), 命名约定, 295
标准清除, st_clean 脚本, 91

表

表, gsscred, 473–474

病

病毒
 拒绝服务攻击, 44
 特洛伊木马, 43
病毒扫描
 描述, 68
 配置, 68–71

病毒扫描（续）

- 文件, 67-68
- 引擎, 67-68

测

- 测试清单, 94

策

策略

- Oracle Solaris 中的定义, 32
- SEAM Tool 面板, 437-440
- 查看列表, 429-431
- 查看属性, 431-433
- 创建 (Kerberos), 421
- 概述, 32
- 管理, 411-446
- 加密框架中的定义, 202
- 口令和, 451
- 删除, 436
- 审计, 496-498
- 新建 (Kerberos), 433-434
- 修改, 435-436
- 用于管理的任务列表, 429
- 在设备上, 74
- 指定口令算法, 59-61

层

层次化领域

- 配置, 341-342
- 在 Kerberos 中, 306, 314-315

插

- 插槽, 加密框架中的定义, 202

插件

- SASL 和, 266
- 从 KMF 中删除, 243-244
- 加密框架, 201

插件（续）

- 审计, 482
- 添加到 KMF, 243-244
- 在 KMF 中管理, 232
- 插入记号 (^), 审计类前缀修饰符, 557
- 插入记号 (^) 前缀, 在 `audit_flags` 值中使用, 506-507

查

查看

- RBAC 权限, 147-149
- shell 中的特权, 177, 180
- XML 审计记录, 535
- 策略列表, 429-431
- 策略属性, 431-433
- 二进制审计文件, 535-537
- 加密机制
 - 可用, 220, 225
 - 目的, 221
 - 现有, 219, 220, 225
- 加密机制的详细列表, 221
- 进程的特权, 180
- 可用加密机制, 220, 225
- 没有口令的用户, 56
- 票证, 448-450
- 权限配置文件的内容, 187
- 设备策略, 74
- 设备分配信息, 78-79
- 审计记录定义, 530-532
- 使用 `list` 命令查看密钥列表缓冲区, 444, 445
- 特权, 175-176
- 特权定义, 176
- 文件的 MAC, 214
- 文件权限, 117-118
- 文件摘要, 212
- 现有加密机制, 220, 225
- 用户的登录状态, 55
- 直接指定的特权, 177
- 主体的属性, 419-421
- 主体列表, 417-419

超

超级用户

- 监视访问尝试, 62–63
- 监视和限制, 61–63
- 解决使 root 成为角色过程中出现的问题, 175
- 特权模型的差别, 139
- 与 RBAC 模型比较, 127–130
- 与特权模型进行比较, 138–144
- 远程访问故障排除, 63
- 在 RBAC 中消除, 136

成

- 成本控制, 和审计, 498
- 成功, 审计类前缀, 557

承

承担角色

- root, 150
- 如何, 153–165
- 在终端窗口中, 149–150

程

程序

- 检查 RBAC 授权, 162
- 可识别特权, 141, 142

初

- 初始票证, 定义, 465

处

- 处理时间成本, 审计服务, 498

窗

- 窗口检验器, 250

创

创建

- root 用户, 173–175
- 安全 Shell 密钥, 277–279
- 存储文件, 340, 381
- 二进制审计文件的存储, 515–518
- 角色, 155–157
- 密钥对, 238–242
- 凭证表, 348
- 权限配置文件, 159–161
- 审计迹, 560
- 使用 kinit 创建票证, 448
- 私钥
 - 用于加密, 206–208, 208–212
- 特权用户, 169
- 文件摘要, 212–213
- 新策略 (Kerberos), 421, 433–434
- 新设备清除脚本, 90–91
- 新主体 (Kerberos), 421–423
- 用户组的权限配置文件, 508

磁

磁带机

- 分配, 82
- 清除数据, 90
- 设备清除脚本, 89

磁带清除脚本, 磁带机, 90

磁盘空间, 二进制审计文件, 515–518

磁盘空间要求, 审计文件, 499

磁盘驱动器, 设备清除脚本, 90

从

从 KDC

- 定义, 464
- 规划为, 316
- 交互配置, 336

从 KDC (续)

- 主 KDC 和, 307
- 自动配置, 335–336

从 KDC 服务器

- 或主, 322
- 交换主 KDC 服务器, 368–372
- 配置, 337–340

存

存储

- 口令短语, 216
- 审计文件, 493, 515–518
- 存储成本, 和审计, 499
- 存储文件
 - 创建, 340, 381
 - 定义, 464
- 存储溢出防止, 审计迹, 538–539

错

- 错误, 分配错误状态, 87
- 错误消息

- encrypt 命令, 217
- Kerberos, 393–406
- 使用 kpasswd, 452

打

- 打印, 审计日志, 536

代

- 代理票证, 定义, 466
- 代理守护进程, 安全 Shell, 280–281

单

- 单点登录系统, 455–460
 - Kerberos 和, 301

登

登录

- ftp 文件传输, 551–552
 - root 登录
 - 跟踪, 42
 - 限于控制台, 62–63
 - 安全
 - 保存失败的尝试, 57
 - 安全性
 - 访问限制, 36
 - 跟踪 root 登录, 42
 - 设备上的访问控制, 40
 - 系统访问控制, 36
 - 和 AUTH_DH, 249
 - 监视故障, 57
 - 临时禁止, 56–57
 - 任务列表, 54
 - 审计登录, 550–551
 - 失败的登录的日志, 57–59
 - 使用安全 Shell, 279–280
 - 显示用户的登录状态, 55
 - 用户的基本特权集, 141
- ### 登录访问限制,
- svc:/system/name-service/switch:default, 36

等

- 等号 (=), 文件权限符号, 114

点

点(.)

- 授权名称分隔符, 188
- 显示隐藏文件, 117

调

- 调试, 特权, 181
- 调试序列号, 568

定

定制, 清单, 98-99
定制报告 (BART), 104-105

读

读权限, 符号模式, 114

端

端口, 用于 Kerberos KDC, 316
端口转发
 安全 Shell, 282
 在安全 Shell 中配置, 275-276

对

对话密钥
 在安全 RPC 模式下解密, 250
 在安全 RPC 中生成, 249
对提供者签名, 加密框架, 204
对象重用要求
 设备清除脚本
 编写新脚本, 90-91
 磁带机, 89
 针对设备, 89-91

范

范围 (RBAC), 说明, 137

方

方括号 ([]), auditrecord 输出, 561

防

防病毒软件, 请参见病毒扫描

防火墙系统

安全性, 49-50
安全主机连接, 283
包粉碎, 50
包传送, 50
从外部连接, 285
可信主机, 50
使用安全 Shell 的外部连接
 从命令行, 285
 从配置文件, 283-285

防止

32 位可执行文件危及安全, 116
审计迹溢出, 538-539
硬件机制的使用, 228-230

访

访问

root 访问
 监视 su 命令尝试, 61-62
 监视 su 命令意图, 42
 限制, 47, 62-63
 在控制台中显示尝试, 62-63

安全

保存失败的登录, 57
系统硬件, 63-65

安全 RPC 验证, 247

安全性

ACL, 46
NFS 客户机-服务器, 248-251
PATH 变量设置, 43
root 登录跟踪, 42
setuid 程序, 44
UFS ACL, 115-116
报告问题, 51
登录访问限制, 36
登录控制, 36
登录验证, 280-281
防火墙设置, 49
监视系统使用情况, 45
控制系统使用情况, 42-45
设备, 74
外围设备, 40
网络控制, 47-50

访问, 安全性 (续)

文件访问限制, 44

物理安全性, 36

远程系统, 269

共享文件, 46

控制列表

请参见ACL

使用安全 Shell 进行登录验证, 280–281

授予您的帐户, 453–455

限制

设备, 40–42, 74

系统硬件, 63–65

限制对 KDC 服务器, 389–390

访问控制列表

请参见ACL访问控制列表 (Access Control List, ACL), **请参见**ACL

访问权

安全性

防火墙设置, 49–50

访问权限

到达服务器

使用 Kerberos, 469–471

为特定服务获取, 471

非

非层次化领域, 在 Kerberos 中, 306

分

分号 (;), device_allocate 文件, 88

分配错误状态, 87

分配多个设备, 故障排除, 82

分配设备

按用户, 81–82

强制, 79

服

服务

Kerberos 中的定义, 464

为特定服务获取访问权限, 471

服务 (续)

在主机上禁用, 445–446

服务管理工具

刷新加密框架, 222

重新启动安全 Shell, 276

服务管理工具 (Service Management Facility, SMF), **请****参见**SMF

服务密钥

Kerberos 中的定义, 465

密钥表文件和, 441–446

服务器

AUTH_DH 客户机-服务器会话, 248–251

Kerberos 中的定义, 464

获取凭证, 470

领域和, 307

使用 Kerberos 获取访问权限, 469–471

针对安全 Shell 配置, 289

服务主体

从密钥表文件删除, 443–444

说明, 306

添加至密钥表文件, 441, 442–443

为名称规划, 315–316

符

符号链接, 文件权限, 111

符号模式

更改文件权限, 114, 119–120, 120

说明, 113

复

复制

文件, 使用安全 Shell, 283

主体 (Kerberos), 424

附

附加箭头 (>>), 防止附加, 43

高

高速缓存, 凭证, 468

更

更改

- audit_class 文件, 513
- audit_event 文件, 514
- NFS 私钥, 249
- root 角色作为用户, 173-175
- root 口令, 54
- 安全 Shell 的口令短语, 279
- 角色的口令, 166-167
- 角色的属性, 167-168
- 可分配设备, 80-81
- 口令算法任务列表, 59-61
- 权限配置文件的内容, 159-161
- 缺省口令算法, 59-61
- 设备策略, 75
- 审计缺省值, 504-505
- 使用 kpasswd 更改您的口令, 451
- 使用 passwd 更改您的口令, 451
- 特殊文件权限, 121-122
- 文件的组所有权, 119
- 文件权限
 - 符号模式, 119-120
 - 绝对模式, 120-121
 - 特殊, 121-122
- 文件所有权, 118-119
- 域的口令算法, 60

公

公共对象, 审计, 480

公共目录

- sticky 位和, 112
- 审计, 480

公钥

- DH 验证和, 248-251
- 安全 Shell 标识文件, 295
- 更改口令短语, 279
- 生成公钥/私钥对, 277-279

公钥技术, 请参见PKI

公钥加密

- AUTH_DH 客户机-服务器会话, 248-251
- NFS 密钥, 249
- 安全 RPC 的公钥数据库, 249
- 更改 NFS 公钥和密钥, 249
- 公用密钥
 - 计算, 250
- 生成密钥
 - 安全 NFS 的对话密钥, 249
 - 使用 Diffie-Hellman, 249
- 公钥验证, 安全 Shell, 270
- 公用密钥
 - DH 验证和, 248-251
 - 计算, 250

共

共享文件

- 和网络安全性, 46
- 通过 DH 验证, 254

故

故障排除

- encrypt 命令, 217
- Kerberos, 406
- list_devices 命令, 79
- praudit 命令, 537
- root 作为角色, 175
- 安全属性, 162-165
- 查找文件setuid 权限, 123
- 发出 su 命令的终端, 62
- 分配一个设备, 82
- 挂载设备, 83
- 活动插件, 541
- 计算机侵入尝试, 57
- 缺少特权, 181-183
- 审计, 539-540
- 审计类
 - 定制的, 513, 541
- 特权要求, 181-183
- 远程超级用户访问, 63
- 运行特权命令的用户, 178-179

故障排除（续）

- 阻止程序使用可执行栈，124

挂

挂载

- 分配的 CD-ROM，83
- 分配的设备，82–83
- 文件通过 DH 验证，254

关

关键字

- 另请参见特定关键字
- BART 中的属性，107
- 安全 Shell，290–295
- 安全 Shell 中的命令行覆盖，297

管

管理

- 另请参见管理

Kerberos

- 策略，429–436
- 密钥表，441–446
- 主体，416–428

metaslot，203

- NFS 客户机-服务器文件安全性，248–251

- RBAC 任务列表，165–166

- RBAC 属性，159–161

- 安全 RPC 任务列表，251

安全 Shell

- 服务器，289
- 概述，287–289
- 客户机，289
- 任务列表，273

安全属性

- 角色，166–167
- 角色的，167–168, 172–173
- 权限配置文件的，159–161
- 用户的，168–169
- 传统应用程序的，161–162

管理（续）

- 不具有特权，139

- 加密框架和区域，204

- 加密框架命令，203

- 加密框架任务列表，218

- 角色口令，166–167

- 口令算法，59–61

- 权限配置文件，159–161

- 用户的，173

- 设备，77

- 设备策略，74

- 设备分配，77

- 设备分配任务列表，77

- 审计，501

- audit -s 命令，526–528, 529

- audit -t 命令，528–529

- audit_remote 插件，520–521

- audit_syslog 插件，521–523

- auditconfig 命令，502–503, 504–505

- auditreduce 命令，532–533

- praudit 命令，535–537

- 在区域中，487–488, 556

- 策略，508–511

- 插件，520–521

- 成本控制，498

- 队列控制，511–512

- 减少空间要求，499

- 禁用，528–529

- 配置，502–503

- 启用，529

- 区域，523–526

- 任务列表，501

- 审计记录，481

- 审计迹溢出防止，538–539

- 审计类，481

- 审计事件，480

- 审计文件，535–537

- 刷新，526–528

- 说明，487

- 所需的权限配置文件，555–556

- 效率，499

- 审计记录任务列表，530

- 审计迹溢出，538–539

- 审计文件，532–533, 538–539

管理 (续)

- 使用 Kerberos 口令, 450–455
- 使用安全 Shell 远程登录, 277–279
- 特权, 179
- 特权任务列表, 179
- 替换超级用户的角色, 153–155
- 文件权限, 116–124
- 用户口令以承担角色, 172–173
- 用户口令以使用权限配置文件, 173
- 由 KMF 管理密钥库, 233
- 在区域中审计, 487–488, 492–493, 556

管理员

- 限制权限, 171–172
- 限制用户的权限, 170–171

归

- 归档, 审计文件, 538–539
- 归档磁带机设备清除脚本, 89

规

规划

Kerberos

- 从 KDC, 316
- 端口, 316
- 客户机与服务主体名称, 315–316
- 领域, 314–315
- 领域分层结构, 314–315
- 领域名称, 314
- 领域数量, 314
- 配置决策, 313–320
- 时钟同步, 318
- 数据库传播, 317

PAM, 257

RBAC, 153–155

审计, 491–496

审计任务列表, 491–496

在区域中审计, 492–493

国

- 国际协调时间 (Coordinated Universal Time, UTC), 在审计中使用时间戳, 560

合

- 合并, 二进制审计记录, 532–533
- 合并审计文件
 - auditreduce 命令, 532–533
 - 从不同区域, 556

环

环境变量

另请参见变量

PATH, 43

安全 Shell, 294–295

存在于审计记录, 496

存在于审计记录中, 561

覆盖代理服务器和端口, 284

审计标记, 564

与 ssh-agent 命令一起使用, 297

恢

- 恢复, 加密提供者, 225

会

会话 ID, 审计, 559

会话密钥

Kerberos 验证和, 468

Kerberos 中的定义, 465

活

- 活动审计策略, 临时审计策略, 508–511

获

获得, 特权, 142

获取

- TGS 的凭证, 469
- 对特定服务的访问权限, 471
- 服务器凭证, 470
- 进程的特权, 180–181
- 可转发票证, 448
- 使用 kinit 获取票证, 448
- 特权, 142, 168, 169
- 特权命令, 167–168

基

基本审计报告工具, **请参见**BART

基本特权集, 141

基于 Kerberos 的命令的选项, 456

基于角色的访问控制, **请参见**RBAC

基于主机的验证

- 说明, 270
- 在安全 Shell 中配置, 273–275

机

机制

- 加密框架中的定义, 201
- 禁用硬件提供者上的所有, 228–230
- 启用硬件提供者上的某些, 229

计

计算

- DH 密钥, 252
- 密钥, 208–212
- 私钥, 206–208
- 文件的 MAC, 213–215
- 文件摘要, 212–213

计算机安全

请参见系统安全

计算机应急响应小组/协调中心 (CERT/CC), 51

加

加号 (+)

- su_{log} 文件中的项, 62
- 审计类前缀, 557
- 文件权限符号, 114

加密

- DES 算法, 248
- encrypt 命令, 215–218
- NIS 用户的私钥, 253
- 安全 NFS, 248
- 保密性服务, 301
- 口令, 59–61
- 口令算法, 37
- 口令算法列表, 37
- 类型

Kerberos 和, 320, 472–473

模式

Kerberos 和, 320

生成对称密钥

- 使用 dd 命令, 206–208
- 使用 pktool 命令, 208–212

使用 -x 选项, 456

使用用户级命令, 203

算法

Kerberos 和, 320

文件, 46, 205–206, 215–218

在 policy.conf 文件中指定口令算法, 37

在 ssh_config 文件中指定算法, 290

指定口令算法

本地, 59–61

主机间的网络通信, 269–271

主机间通信, 280

加密服务, **请参见**加密框架

加密框架

- cryptoadm 命令, 202, 203
- elfsign 命令, 203
- PKCS #11 库, 201
- 错误消息, 217
- 对提供者签名, 204
- 交互, 202
- 连接提供者, 204
- 列出提供者, 218–222
- 区域和, 204, 230
- 任务列表, 205

加密框架（续）

- 使用者, 201
- 刷新, 230
- 说明, 199–201
- 提供者, 201
- 通过角色管理, 158
- 硬件插件, 201
- 用户级命令, 203
- 重新启动, 230
- 术语定义, 201
- 注册提供者, 204

监

监视

- su 命令尝试, 61–62
- su 命令意图, 42
- 超级用户, 61–63
- 超级用户访问尝试, 62–63
- 失败的登录, 57
- 实时审计迹, 499
- 特权命令的使用, 159
- 系统使用情况, 45

减

减号 (-)

- su_{log} 文件中的项, 62
- 审计类前缀, 557
- 文件类型符号, 110
- 文件权限符号, 114

减少

- 审计文件, 532–533
- 审计文件的存储空间要求, 499
- 审计文件所需的磁盘空间, 549–550

检

检验器

- 窗口, 250
- 说明, 250

将

- 将审计记录复制到单个文件, 535

交

交互配置

Kerberos

- 从 KDC 服务器, 336
- 主 KDC 服务器, 324–325

- 交换主与从 KDC 服务器, 368–372

脚

脚本

- audit_warn 脚本, 512–513, 555
- 保护, 162
- 处理 praudit 输出, 536
- 监视审计文件示例, 500
- 检查 RBAC 授权, 162
- 设备清除脚本
 - 另请参见设备清除脚本
- 使用特权, 183
- 使用特权运行, 143
- 用于清除设备, 89–91

角

角色

- 承担, 149–150
- 承担 root 角色, 150
- 创建, 155–157
 - Crypto Management（加密管理）角色, 158
- 概要, 130
- 更改口令, 166–167
- 更改属性, 167–168
- 建议的角色, 127
- 将 root 角色更改为用户, 173–175
- 将特权指定给, 168
- 列出本地角色, 149, 192
- 确定角色的特权命令, 178–179
- 确定直接指定的特权, 177
- 审计, 159

角色 (续)

- 使用 `usermod` 命令进行指定, 157–158
- 使用用户的口令进行验证, 172–173
- 使用用户口令, 132
- 使用指定的角色, 149–150
- 说明, 135–136
- 添加到用户, 168–169
- 修改, 167–168
- 用于访问硬件, 64
- 在 RBAC 中使用, 127
- 在登录后承担角色, 136
- 在终端窗口中承担, 136, 149–150

解

解除分配

- 麦克风, 84
- 强制, 79–80
- 设备, 84

解密

- NFS 密钥, 249
- 安全 RPC 的对话密钥, 250
- 密钥, 249
- 文件, 216

禁

禁用

- 加密机制, 223
- 键盘关闭, 64–65
- 键盘中止, 64–65
- 可执行栈, 124
- 软件提供者
 - 临时, 226
 - 永久, 226, 227
- 设备分配, 78
- 审计策略, 508–511
- 审计服务, 528–529
- 危及安全的 32 位可执行文件, 116
- 系统异常中止序列, 64–65
- 异常中止序列, 64–65
- 硬件机制, 228–230
- 在主机上禁用服务 (Kerberos), 445–446

禁止

- 程序使用可执行栈, 124
- 记录可执行栈消息, 124
- 临时登录, 56–57
- 内核软件提供者使用, 225–227
- 用户登录, 56–57
- 远程 root 访问, 62–63

进

- 进程权限管理, **请参见** 特权
- 进程审计特征
 - 进程预选掩码, 559
 - 审计会话 ID, 559
 - 审计用户 ID, 559
 - 终端 ID, 560
- 进程特权, 139
- 进程预选掩码, 说明, 559

井

井号 (#)

- `device_allocate` 文件, 89
- `device_maps` 文件, 88

绝

绝对模式

- 更改特殊文件权限, 121–122
- 更改文件权限, 113, 120–121
- 设置特殊权限, 114
- 说明, 113

开

- 开始, 设备分配, 77–78

可

- 可插拔验证模块, **请参见** PAM

- 可代理票证, 定义, 466
- 可读审计记录格式, 将审计记录转换为, 536
- 可更新票证, 定义, 466
- 可继承特权集, 141
- 可信主机, 50
- 可执行栈
 - 禁止记录消息, 124
 - 日志记录消息, 116
 - 阻止, 124
 - 阻止 32 位进程, 116
- 可转发票证
 - 定义, 465
 - 使用 -F 选项, 456, 457-458
 - 使用 -f 选项, 456, 457-458
 - 示例, 448
 - 说明, 302

客

- 客户机
 - AUTH_DH 客户机-服务器会话, 248-251
 - Kerberos 中的定义, 464
 - 配置 Kerberos, 351-367
 - 为安全 Shell 配置, 288
 - 针对安全 Shell 配置, 289
- 客户机名称, 在 Kerberos 中规划, 315-316

控

- 控制
 - 系统访问, 53-54
 - 系统使用情况, 42-45
- 控制清单 (BART), 93
- 控制台, 显示 su 命令尝试, 62-63

口

- 口令
 - LDAP, 37
 - 指定新的口令算法, 60-61
 - NIS, 37
 - 指定新的口令算法, 60

口令 (续)

- PROM 安全模式, 36, 63-65
- UNIX 和 Kerberos, 450-455
- 安全 RPC 的密钥解密, 249
- 保护
 - PKCS #12 文件, 237
 - 密钥库, 237
- 本地, 37
- 策略和, 451
- 查找没有口令的用户, 56
- 登录安全性, 36
- 更改角色的口令, 166-167
- 关于选择的建议, 450-451
- 管理, 450-455
- 加密算法, 37
- 将 MD5 加密算法用于, 59-60
- 任务列表, 54
- 使用 kpasswd 命令进行更改, 451
- 使用 passwd -r 命令更改, 37
- 使用 passwd 命令进行更改, 451
- 使用新的算法, 59
- 使用用户的口令承担角色, 172-173
- 授予访问权限但不显示, 453-455
- 系统登录, 36
- 显示没有口令的用户, 56
- 修改主体的口令, 425
- 要求硬件访问, 64
- 硬件访问和, 64
- 在安全 Shell 中消除, 280-281
- 在安全 Shell 中验证, 270
- 在异构环境中使用 Blowfish, 60
- 在异构环境中限制加密算法, 60
- 指定算法, 59-60
 - 本地, 59-61
 - 在命名服务中, 60

口令短语

- encrypt 命令, 215
- mac 命令, 213
- 安全存储, 216
- 示例, 280
- 用于 MAC, 214
- 在 KMF 中生成, 238
- 在安全 Shell 中使用, 280-281
- 针对安全 Shell 更改, 279

口令验证, 安全 Shell, 270

库

库, 用户级提供者, 219

跨

跨领域验证, 配置, 341–343

类

类, 请参见 审计类

联

联机帮助

SEAM Tool, 413

URL, 320

联机帮助 URL, Kerberos 图形工具, 320

列

列表特权, SEAM Tool 和, 440

列出

RBAC 权限, 147–149

加密框架提供者, 227–228

加密框架中的可用提供者, 218–222

加密框架中的提供者, 218–222

没有口令的用户, 56

密钥库内容, 234

您可以承担的角色, 149, 192

设备策略, 74

所有 RBAC 安全属性, 146–147

硬件提供者, 227–228

临

临时审计策略

活动审计策略, 508–511

设置, 510

领

领域 (Kerberos)

层次化, 341–342

层次化或非层次化, 306

分层结构, 314–315

服务器和, 307

将主机名映射到, 315

名称, 314

内容, 307

配置决策, 314–315

配置跨领域验证, 341–343

请求特定票证, 456

数目, 314

在主体名称中, 305–306

直接, 342–343

令

令牌, 加密框架中的定义, 202

麦

麦克风

分配, 81–82

解除分配, 84

密

密钥

Kerberos 中的定义, 465

创建, 208–212

服务密钥, 441–446

会话密钥

Kerberos 验证和, 468

密钥 (续)

生成对称密钥

使用 dd 命令, 206–208

使用 pktool 命令, 208–212

生成密钥对

使用 pktool 命令, 238–242

为 NIS 用户创建 DH 密钥, 253–254

为安全 RPC 生成, 249

为安全 Shell 创建, 277–279

为安全 Shell 生成, 277–279

用于 MAC, 214

密钥表文件

管理, 441–446

将服务主体添加至, 441, 442–443

将主 KDC 服务器的 host 主体添加到, 328, 334

删除服务主体, 443–444

使用 delete_entry 命令禁用主机的服务, 445

使用 ktremove 命令删除主体, 443

使用 ktutil 命令查看内容, 443, 444–445

使用 ktutil 命令进行管理, 441

使用 list 命令查看密钥列表缓冲区, 444, 445

使用 read_kt 命令读入密钥表缓冲区, 444, 445

密钥对

创建, 238–242

生成

使用 pktool 命令, 238–242

密钥分发中心, 请参见 KDC**密钥管理框架 (key management framework, KMF), 请参见 KMF****密钥库**

导出证书, 237–238

导入证书, 235–236

列出内容, 234

受 KMF 支持, 231, 233

由 KMF 管理, 232

在 KMF 中使用口令保护, 238

面

面板, SEAM Tool 表, 437–440

名

名称

设备名称

device_maps 文件, 88, 89

命

命令

另请参见单个命令

Kerberos, 462–463

RBAC 管理命令, 192

安全 RPC 命令, 248

安全 Shell 命令, 297–298

加密框架命令, 203

检查特权, 134

确定用户的特权命令, 178–179

设备策略命令, 85

设备分配命令, 86

文件保护命令, 109

用户级加密命令, 203

用于管理特权, 194

指定特权, 142

命令执行, 安全 Shell, 289

命名服务

请参见单个命名服务

范围和 RBAC, 137

命名服务配置, 登录访问限制, 36

命名约定

RBAC 授权, 188

安全 Shell 标识文件, 295

设备, 79

审计文件, 560

模

模块, 口令加密, 37

模式, 加密框架中的定义, 202

目

目录

另请参见文件

目录 (续)

- 公共目录, 112
- 权限
 - 缺省值, 112–113
 - 说明, 110–111
- 显示文件和相关信息, 109, 117–118

内

- 内核提供者, 列出, 219

配

配置

- ahlt 审计策略, 510
- audit_class 文件, 513
- audit_event 文件, 514
- audit_warn 脚本, 512–513
- Kerberos
 - NFS 服务器, 346–348
 - 从 KDC 服务器, 335–336, 336, 337–340
 - 概述, 321–391
 - 客户机, 351–367
 - 跨领域验证, 341–343
 - 任务列表, 321–322
 - 使用 LDAP 的主 KDC 服务器, 329–335
 - 添加管理主体, 327, 333
 - 主 KDC 服务器, 323–324, 324–325, 325–329
- NIS 用户的 DH 密钥, 253–254
- NIS 中的 DH 密钥, 252–253
- perzone 审计策略, 510–511
- RBAC, 153–165
- RBAC 任务列表, 153
- root 角色作为用户, 173–175
- 安全 Shell, 272–273
 - 服务器, 289
 - 客户机, 289
- 安全 Shell 的基于主机的验证, 273–275
- 安全 Shell 任务列表, 273
- 安全 Shell 系统缺省值例外, 276–277
- 安全 Shell 中端口转发, 275–276
- 活动审计策略, 510
- 角色, 155–157, 167–168

配置 (续)

- 临时审计策略, 508–511
- 每区域审计, 525–526
- 权限配置文件, 159–161
- 设备策略, 74
- 设备分配, 77
- 设备任务列表, 73
- 审计, 502–514
- 审计策略, 508–511
- 审计队列控制, 511–512
- 审计服务策略, 508–511
- 审计记录的文本摘要, 521–523
- 审计迹的空间, 518–520
- 审计迹溢出防止, 538–539
- 审计类, 504–505
- 审计任务列表, 502–503
- 审计日志任务列表, 515
- 特权用户, 169
- 以相同方式审计非全局区域, 523–525
- 硬件安全, 63–65
- 永久审计策略, 508–511
- 用于访问硬件的口令, 64
- 在区域中审计, 487–488, 556
- 配置的审计策略, 永久审计策略, 508–511
- 配置决策
 - Kerberos
 - KDC 服务器, 319
 - 从 KDC, 316
 - 端口, 316
 - 加密类型, 320
 - 将主机名映射到领域, 315
 - 客户机, 318
 - 客户机与服务主体名称, 315–316
 - 领域, 314–315
 - 领域分层结构, 314–315
 - 领域名称, 314
 - 领域数量, 314
 - 时钟同步, 318
 - 数据库传播, 317
 - 口令算法, 37
 - 审计
 - 策略, 496–498
 - 区域, 492–493
 - 文件存储, 493

配置决策, 审计 (续)

要审计的对象及内容, 494–496

配置文件

请参见权限配置文件

device_maps 文件, 87

用于口令算法, 37

policy.conf 文件, 37, 59–60, 192

syslog.conf 文件, 57–59, 195

安全 Shell, 287

包含特权信息, 194–195

审计, 554–555

配置文件 shell

打开, 150–152

将用户限于桌面应用程序, 170–171

说明, 136

限制权限, 171–172

配置应用服务器, 343–345

票

票证

-F 选项或 -f 选项, 456

-k 选项, 456

Kerberos 中的定义, 465

klist 命令, 448–450

查看, 448–450

初始, 465

创建, 447

代理, 466

定义, 302

或凭证, 302

获取, 447

可代理, 466

可更新, 466

可以后生效, 466

可转发, 302, 448, 457–458, 465

类型, 465–468

生命周期, 466–467

失效警告, 361

使用 kinit 创建, 448

文件

请参见凭证高速缓存

无效, 466

销毁, 450

票证 (续)

以后生效的, 302

针对特定领域而请求, 456

最长可更新生命周期, 467

票证的生命周期, 在 Kerberos 中, 466–467

票证类型, 465–468

票证失效警告, 361

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, 468

或票证, 302

说明, 249, 465

为 TGS 获取, 469

为服务器获取, 470

映射, 317

凭证表, 添加单个项, 348–349

启

启动

KDC 守护进程, 340, 381

安全 RPC keyserver, 251–252

审计, 529

启用

加密机制, 224

键盘中止, 64–65

仅基于 Kerberos 的应用程序, 389

内核软件提供者使用, 225

设备分配, 77–78

审计, 529

审计服务, 529

硬件提供者的机制和功能, 229

签

签署

PKCS #10 CSR, 242–243

签署, PKCS #10 CSR (续)

使用 pktool 命令, 242–243

强

强制清除, st_clean 脚本, 91

清

清除, 二进制审计文件, 537–538

清单

另请参见 bart create

测试, 94

定制, 98–99

控制, 93

文件格式, 105–106

区

区域

perzone 审计策略, 487–488, 493, 556

zonename 审计策略, 493, 556

规划审计, 492–493

加密服务和, 230

加密框架和, 204

设备和, 41

审计和, 487–488, 556

在全局区域中配置审计, 510

取

取消挂载, 分配的设备, 84

全

全限定域名 (Fully Qualified Domain Name, FQDN),

在 Kerberos 中, 315–316

权

权限

请参见 权限配置文件

ACL 和, 46

setgid 权限

符号模式, 114

绝对模式, 114, 122

说明, 112

setuid 权限

安全风险, 112

符号模式, 114

绝对模式, 114, 122

说明, 111–112

sticky 位, 112

UFS ACL 和, 115–116

umask 值, 112–113

查找文件 setuid 权限, 123

更改文件权限

chmod 命令, 110

符号模式, 113, 114, 119–120, 120

绝对模式, 113, 120–121

将管理员限于显式指定的, 171–172

将用户限于桌面应用程序, 170–171

目录权限, 110–111

缺省值, 112–113

特殊文件权限, 111–112, 112, 114

文件权限

符号模式, 113, 114, 119–120, 120

更改, 113–115, 120

绝对模式, 113, 120–121

说明, 110–111

特殊权限, 112, 114

用户类和, 110

权限配置文件

All (所有), 186

审计服务, 555–556

Basic Solaris User (基本 Solaris 用户), 186

Console User (控制台用户), 186, 187

Operator (操作员), 186

Printer Management (打印机管理), 186

Stop (停止), 186, 187

System Administrator (系统管理员), 186

查看内容, 187

典型内容, 185

权限配置文件（续）

- 防止特权升级, 129, 155
- 更改内容, 159–161
- 故障排除, 162–165
- 设备安全, 77–78, 86
- 设备管理, 86
- 使用 System Administrator（系统管理员）配置文件, 64
- 使用用户的口令进行验证, 173
- 数据库
 - 请参见 prof_attr 数据库和 exec_attr 数据库
- 说明, 130, 135
- 搜索顺序, 187
- 修改, 159–161
- 指定给可信用户, 129, 155
- 主要权限配置文件说明, 185

缺

缺省安全安装选项, 44

缺省值

- policy.conf 文件中的特权设置, 194
- policy.conf 文件中的系统范围, 37
- umask 值, 112–113
- 审计服务, 553–554

确

确定

- 进程的特权, 180–181
- 审计正在运行, 540–542
- 特权任务列表, 175
- 文件 setuid 权限, 123
- 用户的审计 ID, 547

任

任务列表

- Kerberos 配置, 321–322
- Kerberos 维护, 322
- PAM, 257
- 安全 Shell, 272–273

任务列表（续）

- 保护系统, 53–54
- 保证登录和口令的安全, 54
- 防止程序受到安全风险, 122
- 管理 RBAC, 165–166
- 管理安全 RPC, 251
- 管理策略 (Kerberos), 429
- 管理和使用特权, 175
- 管理加密框架, 218
- 管理设备策略, 74
- 管理设备分配, 77
- 管理审计记录, 530
- 管理主体 (Kerberos), 416
- 规划审计, 491–496
- 加密框架, 205
- 配置 Kerberos NFS 服务器, 346
- 配置 RBAC, 153
- 配置安全 Shell, 273
- 配置设备, 73
- 配置设备策略, 74
- 配置审计, 502–503
- 配置审计日志, 515
- 设备, 73
- 设备策略, 74
- 设备分配, 77
- 审计, 501
- 审计的故障排除, 539–540
- 使用 BART 任务列表, 96–97
- 使用 RBAC, 145–146
- 使用 UNIX 权限保护文件, 117
- 使用安全 Shell, 277
- 使用加密机制保护文件, 205–206
- 使用加密框架, 205
- 使用密钥管理框架（任务列表）, 233–234
- 使用缺省 RBAC 配置, 146
- 系统访问, 53–54

认

认证者, 在 Kerberos 中, 470

日

日志文件

BART

- 程序输出, 108

- 详细输出, 108

- syslog 审计记录, 555

- /var/adm/messages, 541

- /var/log/syslog, 541

- 监视 su 命令, 61–62

- 审计记录, 483, 536

- 失败的登录尝试, 57–59

- 为审计服务配置, 521–523

散

散列

算法

- Kerberos 和, 320

- 文件, 205–206

删

删除

- audit_event 文件中的审计事件, 548–549

- KMF 中的插件, 243–244

- not_terminated 审计文件, 537–538

- 策略 (Kerberos), 436

- 归档的审计文件, 538

- 基本特权集中的特权, 160

- 加密提供者, 224, 225

- 密钥表文件中的服务主体, 443–444

- 设备策略, 75

- 设备的策略, 75

- 审计文件, 532

- 使用 ktremove 命令删除主体, 443

- 限制特权集中的特权, 160, 169

- 用户特定的审计, 508

- 主机的服务, 445

- 主体 (Kerberos), 425–426

上

- 上下文有关帮助, SEAM Tool, 413

设

设备

- /dev/urandom 设备, 206–208

- 安全性, 40–42

- 不需要授权使用, 80

- 策略命令, 85

- 查看分配信息, 78–79

- 查看设备策略, 74

- 超级用户模型和, 144

- 登录访问控制, 40

- 分配以使用, 76–81

- 更改哪些是可分配的, 80–81

- 更改设备策略, 75

- 挂载分配的设备, 82–83

- 管理, 74

- 管理分配, 77

- 获取 IP MIB-II 信息, 76

- 解除分配设备, 84

- 禁止使用某些, 80

- 禁止使用所有, 80–81

- 列出, 74

- 列出设备名称, 79

- 强制分配, 79

- 强制解除分配, 79–80

- 区域和, 41

- 取消挂载分配的设备, 84

- 删除策略, 75

- 设备分配

- 请参见设备分配

- 审计策略更改, 76

- 审计分配, 81

- 使可分配, 77–78

- 授权用户分配, 78

- 特权模型和, 144

- 添加设备策略, 75

- 通过设备分配保护, 40

- 在内核中保护, 40

- 设备安全权限配置文件, 77–78, 86

- 设备策略

- add_drv 命令, 85

设备策略 (续)

- update_drv 命令, 75, 85
- 查看, 74
- 从设备中删除, 75
- 概述, 40-42
- 更改, 75
- 管理设备, 74
- 命令, 85
- 内核保护, 84-91
- 配置, 74-76
- 任务列表, 74
- 审计更改, 76

设备分配

- deallocate 命令
 - 设备清除脚本和, 90-91
 - 使用, 84
- device_allocate 文件, 88-89
- device_maps 文件, 87-88
- SMF 服务, 86
- 不需要授权, 80
- 查看信息, 78-79
- 分配错误状态, 87
- 分配设备, 81-82
- 更改可分配设备, 80-81
- 故障排除, 82, 83
- 故障排除权限, 79
- 挂载设备, 82-83
- 管理设备, 77
- 机制的组成部分, 85-86
- 解除分配设备, 84
- 禁用, 78
- 可分配设备, 89
- 命令, 86
- 命令的授权, 87
- 配置文件, 87
- 启用, 77-78
- 强制分配设备, 79
- 强制解除分配设备, 79-80
- 取消挂载分配的设备, 84
- 权限配置文件, 86
- 任务列表, 77
- 设备清除脚本
 - CD-ROM 驱动器, 90
 - 编写新脚本, 90-91

设备分配, 设备清除脚本 (续)

- 磁带机, 89, 90
- 磁盘驱动器, 90
- 说明, 89-91
- 选项, 90
- 音频设备, 90

- 审计, 81
- 使设备可分配, 77-78
- 使用, 76-81
- 使用 allocate 命令, 81-82
- 示例, 82
- 授权, 86
- 授权用户分配, 78
- 添加设备, 77
- 要求授权, 80-81
- 用户操作过程, 76-81
- 阻止, 80-81

设备管理, 请参见设备策略

- 设备管理权限配置文件, 86

设备清除脚本

- CD-ROM 驱动器, 90
- 编写新脚本, 90-91
- 磁带机, 89
- 磁盘驱动器, 90
- 和对象重用, 89-91
- 说明, 89-91
- 选项, 90
- 音频设备, 90

设置

- arge 策略, 545
- argv 策略, 545
- 审计策略, 508-511
- 审计队列控制, 511-512
- 主体缺省值 (Kerberos), 426-427

审

审计

- sftp 文件传输, 551-552
- 插件模块, 482
- 当前发行版中的更改, 488-489
- 登录, 550-551
- 更新信息, 526-528
- 故障排除, 539-540

审计 (续)

- 故障排除 `praudit` 命令, 537
- 规划, 491-496
- 后选定义, 479
- 获取队列控制, 511-512
- 角色, 159
- 仅用户, 507-508
- 禁用, 528-529
- 配置
 - 每区域, 525-526
 - 全局区域, 510
 - 所有区域, 502-514
 - 针对所有区域以相同方式, 523-525
- 启用, 529
- 区域和, 487-488, 556
- 权限配置文件, 555-556
- 缺省值, 553-554
- 确定是否正在运行, 540-542
- 删除用户特定的审计标志, 508
- 设备策略的更改, 76
- 设备分配, 81
- 设置队列控制, 511-512
- 手册页摘要, 554-555
- 特权和, 195
- 向用户组添加审计标志, 508
- 用户执行的所有命令, 544-545
- 预选定义, 479
- 在区域中规划, 492-493
- 在全局区域中配置, 492
- 找到对特定文件的更改, 545-547

审计标记

另请参见 各个审计标记名称

- `xcClient` 标记, 570
- 格式, 561
- 列表, 561
- 审计记录格式, 560
- 说明, 479, 482
- 通过审计策略添加, 558

审计标志, 摘要, 478

审计策略

- `public`, 497
- 被添加的标记, 558
- 不影响标记, 558
- 缺省, 496-498

审计策略 (续)

- 设置, 508-511
- 设置 `ahlt`, 510
- 设置 `arge`, 545
- 设置 `perzone`, 510-511
- 设置 `argv`, 545
- 审计标记, 558
- 说明, 479
- 显示缺省值, 503-504
- 影响, 496-498
- 在全局区域设置, 487-488
- 在全局区域中设置, 556

审计插件

- `audit_binfile` 插件, 511-512, 518-520
- `audit_remote` 插件, 520-521
- `audit_syslog` 插件, 521-523
- `qsize` 属性, 511-512
- 说明, 478
- 摘要, 554-555, 557-558

审计队列, 包括的事件, 481

审计队列控制

- 获取, 511-512
- 显示缺省值, 503-504

审计服务

- 另请参见** 审计策略, 496
- 故障排除, 540-542
- 禁用, 528-529
- 配置策略, 508-511
- 配置队列控制, 511-512
- 启用, 529
- 缺省值, 553-554
- 审计迹创建, 560
- 刷新内核, 526
- 审计会话 ID, 559
- 概述, 477-478

审计记录

- `/var/adm/auditlog` 文件, 522
- 标记序列, 560
- 复制到单个文件, 535
- 概述, 481
- 格式, 560
- 格式示例, 530
- 合并, 532-533

审计记录 (续)

减少审计文件, 532-533

生成的事件, 485

事件修饰符, 565

说明, 479

显示, 535-537

显示程序的格式, 531

显示定义

过程, 530-532

显示审计类的格式, 531-532

以 XML 格式显示, 536

转换为可读格式, 536

审计记录的格式, `auditrecord` 命令, 530

审计迹

查看事件, 535-537

创建

摘要文件, 534

从不同区域查看事件, 556

从中选择事件, 533-535

防止溢出, 538-539

分析成本, 498

概述, 486

减小大小, 542-544, 549-550

清除未终止文件, 537-538

审计策略的影响, 496

实时监控, 499

说明, 479

添加磁盘空间, 518-520

无公共对象, 480

向远程系统信息库发送文件, 520-521

审计类

概述, 481

后选, 479

进程预选掩码, 559

配置, 556-557

前缀, 557

说明, 478, 480

替换, 504-505

添加, 513

系统范围的设置的例外, 481

显示缺省值, 503-504

修改缺省, 513

映射事件, 481

用户例外, 505-508

审计类 (续)

语法, 556, 557

预选, 479

成功, 506-507, 521, 522

成功和失败, 504-505

对公共对象的影响, 480

失败, 506-507, 521, 522

审计类前缀, 557

审计类前缀中的 ^ (插入记号), 505-508, 546

审计类前缀中的 + (加号), 521

审计类前缀中的插入记号 (^), 505-508, 546

审计类前缀中的加号 (+), 521

审计目录, 创建文件系统, 515-518

审计任务列表, 501

审计日志

另请参见 审计文件

比较二进制和文本摘要, 483

模式, 483

配置, 515-523

配置文本摘要审计日志, 521-523

审计事件

`audit_event` 文件, 480

从 `audit_event` 文件中删除, 548-549

从二进制文件查看, 535-537

从区域的审计迹中选择, 556

从审计迹中选择, 533-535

更改类成员身份, 514

说明, 480

同步, 558-559

异步, 558-559

映射到类, 481

摘要, 478

审计事件到类映射, 更改, 514

审计特征

会话 ID, 559

进程, 559-560

审计用户 ID, 559

用户进程预选掩码, 559

终端 ID, 560

审计文件

ZFS 文件系统, 515-518, 549-550

创建摘要文件, 534, 535

打印, 536

管理, 538-539

审计文件（续）

- 合并, 532–533
- 减少, 532–533
- 减少存储空间要求, 499
- 减少空间要求, 499
- 将消息复制到单个文件, 535
- 留出磁盘空间, 515–518
- 时间戳, 560
- 使用 praudit 读取, 535–537
- 限制大小, 549
- 协调通用时间 (Coordinated Universal Time, UTC)
 - 的效果, 532
 - 在磁盘上压缩, 549–550
- 审计文件的大小
 - 减少, 532–533
 - 减少存储空间要求, 499
- 审计文件系统, 说明, 478
- 审计用户 ID
 - 概述, 477–478
 - 机制, 559
- 审计预选掩码
 - 为单个用户修改, 505–508
 - 为现有用户修改, 547–548
- 审计中的后选, 479
- 审计中的预选, 479

生

生成

- NFS 密钥, 249
- X.509 v3 证书, 242–243
- 安全 Shell 的密钥, 277–279
- 安全 Shell 密钥, 277–279
- 对称密钥
 - 使用 dd 命令, 206–208
 - 使用 pktool 命令, 208–212
- 口令短语使用 pktool 命令, 238
- 密钥对
 - 使用 pktool 命令, 238–242
- 随机数
 - 使用 dd 命令, 206–208
 - 使用 pktool 命令, 208–212
- 证书通过 pktool 命令, 234–235

失

- 失败, 审计类前缀, 557
- 失败的登录尝试
 - loginlog 文件, 57
 - syslog.conf 文件, 57–59

实

- 实例, 在主体名称中, 305–306

时

- 时间戳, 审计文件, 560
- 时钟同步
 - Kerberos 从 KDC 服务器和, 340
 - Kerberos 从服务器和, 381
 - Kerberos 规划与, 318
 - Kerberos 主 KDC 服务器和, 329, 335
- 时钟相位差
 - Kerberos 规划与, 318
 - Kerberos 和, 367–368

使

- 使得, 设备分配, 81–82
- 使用
 - allocate 命令, 81–82
 - BART, 96
 - cryptoadm 命令, 218
 - dd 命令, 206–208
 - deallocate 命令, 84
 - digest 命令, 212–213
 - encrypt 命令, 215–218
 - mac 命令, 213–215
 - pktool 命令, 208–212, 238–242
 - ppriv 命令, 180
 - RBAC 缺省值, 146–152
 - RBAC 任务列表, 145–146
 - rolemod 命令, 168
 - ssh-add 命令, 280–281
 - ssh-agent 守护进程, 280–281
 - truss 命令, 181

使用 (续)

- umount 命令, 84
- usermod 命令, 169
- 安全 Shell 任务列表, 277
- 加密框架任务列表, 205
- 缺省 RBAC 配置任务列表, 146
- 特权任务列表, 175
- 文件权限, 116–124
- 新的口令算法, 59
- 使用密钥管理框架 (任务列表), 233–234
- 使用者, 加密框架中的定义, 201

事

- 事件, 说明, 480
- 事件修饰符, 审计记录, 565

守

守护进程

- kcfd, 203
- Kerberos 表, 463–464
- keyserv, 251–252
- nscd (名称服务高速缓存守护进程), 192
- ssh-agent, 280–281
- sshd, 287–289
- 使用特权运行, 139

手

- 手册页, 审计服务, 554–555
- 手动配置
 - Kerberos
 - 从 KDC 服务器, 337–340
 - 使用 LDAP 的主 KDC 服务器, 329–335
 - 主 KDC 服务器, 325–329

受

- 受限 shell (rsh), 43

授

授权

- Kerberos 和, 301
- 故障排除, 162–165
- 类型, 48–49
- 设备分配, 86

授权 (RBAC)

- solaris.device.allocate, 78, 87
- solaris.device.revoke, 87
- 定义, 133–134
- 检查通配符, 162
- 粒度, 188
- 命名约定, 188
- 设备分配不需要, 80
- 数据库, 189–191
- 说明, 130, 188–189
- 委托, 188–189
- 要求授权的命令, 192–193
- 用于分配设备, 78
- 用于设备分配, 87
- 在特权应用程序中检查, 135

属

- 属性, BART 中的关键字, 107

数

数据加密标准, 请参见 DES 加密

数据库

- auth_attr, 190
- cred 用于安全 RPC, 248
- exec_attr, 191
- KDC 传播, 317
- NFS 密钥, 249
- prof_attr, 190–191
- publickey 用于安全 RPC, 248
- RBAC, 189–191
- user_attr, 189–190
- 备份和传播 KDC, 372–373
- 创建 KDC, 327
- 数据转发, 安全 Shell, 289

刷

刷新

- 加密服务, 230
- 审计服务, 526–528, 527–528

双

- 双美元符号 (\$\$), 父 shell 进程号, 180

私

私钥

- 另请参见密钥
- Kerberos 中的定义, 465
- 安全 Shell 标识文件, 295
- 创建, 206–208
- 生成
 - 使用 dd 命令, 206–208
 - 使用 pktool 命令, 208–212

搜

搜索顺序

- 安全属性, 187
- 用户安全属性, 187

算

算法

- ssh-keygen 中的口令短语保护, 271
- 加密框架中的定义, 201
- 口令
 - 配置, 59–60
- 口令加密, 37, 59–61
- 文件加密, 215–218
- 在加密框架中列出, 218–222

随

随机数

- dd 命令, 206–208
- pktool 命令, 208–212

特

特洛伊木马, 43

特权

- PRIV_PROC_LOCK_MEMORY, 140
- 保护内核进程, 138
- 查找缺少的, 182
- 超级用户模型的差别, 139
- 从基本特权集中删除, 160
- 从限制特权集中删除, 160, 169
- 从用户删除, 143
- 调试, 144, 181
- 对 SEAM Tool 的影响, 440
- 故障排除
 - 用户, 162–165
- 故障排除要求, 181–183
- 管理, 179
- 具有指定特权的进程, 142
- 可识别特权的程序, 142
- 类别, 139
- 列出, 176
- 命令, 194
- 确定直接指定的特权, 176–177
- 任务列表, 175
- 如何使用, 175
- 设备和, 144
- 审计和, 195
- 升级, 195
- 使用特权执行命令, 143
- 说明, 130, 139
- 添加到命令, 161
- 文件, 194–195
- 由进程继承, 142
- 与超级用户进行比较, 138–144
- 在 shell 脚本中使用, 183
- 在集中实现, 141
- 在进程上列出, 180–181
- 在权限配置文件中限制使用, 160
- 指定给脚本, 143

特权 (续)

- 指定给角色, 168

- 指定给命令, 142

- 指定给用户, 142, 169

特权端口, 安全 RPC 替代项, 49

特权集

- 基本, 141

- 可继承, 141

- 列出, 141

- 删除特权, 143

- 添加特权, 143

- 限制, 141

- 有效, 141

- 允许的, 141

特权检查, 在应用程序中, 134

特权应用程序

- ID 检查, 134

- 授权检查, 135

- 说明, 130

- 特权检查, 134

特殊权限

- setgid 权限, 112

- setuid 权限, 111-112

- sticky 位, 112

提**提供者**

- 恢复内核软件提供者的使用, 225

- 加密框架中的定义, 202

- 禁用硬件机制, 228-230

- 禁止使用内核软件提供者, 225-227

- 连接到加密框架, 204

- 列出硬件提供者, 227-228

- 签名, 204

- 添加库, 223

- 添加软件提供者, 222-223

- 添加用户级软件提供者, 223

- 在加密框架中列出, 218-222

- 注册, 204

- 作为插件的定义, 201

替**替换**

- 使用角色替换超级用户, 153-155

- 预选的审计类, 504-505

添**添加**

- cryptomgt 角色, 158

- PAM 模块, 258

- RBAC 属性

- 到传统应用程序, 161-162

- 安全设备, 76-81

- 安全属性

- 到角色, 167-168

- 到用户, 168-169

- 到传统应用程序, 161-162

插件

- KMF, 243-244

- 加密框架, 222-223

- 审计, 520-521, 521-523

- 对挂载的文件系统进行 DH 验证, 251

- 服务主体到密钥表文件 (Kerberos), 442-443

- 管理主体 (Kerberos), 327, 333

- 角色, 155-157

- 可分配设备, 77-78

- 库插件, 223

- 临时审计策略, 510

- 软件提供者, 222-223

- 设备安全, 75

审计

- 单个用户, 505-508

- 各个用户, 543

- 角色, 159

- 区域, 491-496

- 审计策略, 508-511

- 审计类, 513

- 审计文件系统, 515-518

特权

- 到命令, 161

- 直接到角色, 168

- 直接到用户, 169

- 特权用户, 169

- 系统硬件的安全性, 64

添加 (续)

- 新的权限配置文件, 159–161
- 硬件提供者机制和功能, 229
- 用户级软件提供者, 223
- 与安全相关的角色, 158

通

通配符

- 在 RBAC 授权中, 188
- 针对安全 Shell 中的主机, 284

同

同步时钟

- 从 KDC, 340, 381
- 概述, 367–368
- 主 KDC, 329, 335

透

透明度, Kerberos 中的定义, 302

完

完整性

- Kerberos 和, 301
- 安全服务, 308

网

网关, 请参见防火墙系统

网络, 相关特权, 139

网络安全性

- 报告问题, 51
- 防火墙系统
 - 包粉碎, 50
 - 可信主机, 50
 - 需要, 49
- 概述, 47

网络安全性 (续)

- 控制访问, 47–50
 - 授权, 48–49
 - 验证, 48–49
- 网络时间协议, 请参见 NTP

委

委托, RBAC 授权, 188–189

文

文件

- audit_class, 554
- audit_event, 555
- BART 清单, 105–106
- kdc.conf, 466
- Kerberos, 461–462
- PKCS #12, 237
- syslog.conf, 555
- 安全
 - umask 缺省, 112–113
 - UNIX 权限, 109–115
 - 更改权限, 113–115, 120
 - 目录权限, 110–111
 - 特殊文件权限, 114
 - 文件类型, 110
 - 文件权限, 110–111
 - 显示文件信息, 109, 117–118
 - 用户类, 110

安全性

- ACL, 46
- 访问限制, 44
- 更改所有权, 118–119
- 加密, 46, 205–206
- 包含特权信息, 194–195
- 查找文件 setuid 权限, 123
- 更改所有权, 109, 118–119
- 更改特殊文件权限, 121–122
- 更改组所有权, 119
- 公共对象, 480
- 计算 MAC, 213–215
- 计算摘要, 212–213

文件 (续)

- 加密, 205–206, 215–218
- 解密, 216
- 清单 (BART), 105–106
- 权限
 - setgid, 112
 - setuid, 111–112
 - sticky 位, 112
 - umask 值, 112–113
 - 符号模式, 113, 114, 119–120, 120
 - 更改, 110, 113–115, 120
 - 绝对模式, 113, 120–121
 - 缺省值, 112–113
 - 说明, 110–111
- 散列, 205–206
- 审计修改, 545–547
- 使用 digest 检验完整性, 212–213
- 使用 UNIX 权限保护, 117
- 使用安全 Shell 复制, 283
- 所有权
 - 和 setgid 权限, 112
 - 和 setuid 权限, 111–112
- 特殊文件, 111–112
- 通过 DH 验证共享, 254
- 通过 DH 验证挂载, 254
- 文件类型, 110
- 文件类型的符号, 110
- 显示文件信息, 117–118
- 显示相关信息, 109
- 显示隐藏文件, 117
- 相关特权, 139
- 用于管理安全 Shell, 295
- 摘要, 212–213
- 文件 vnode 审计标记, 563
- 文件的所有权
 - ACL 和, 46
 - UFS ACL 和, 115–116
 - 更改, 109, 118–119
 - 更改组所有权, 119
- 文件的用户类, 110
- 文件权限模式
 - 符号模式, 114
 - 绝对模式, 113

文件系统

- NFS, 247
- TMPFS, 112
- 安全
 - TMPFS 文件系统, 112
- 安全性
 - 验证和 NFS, 247
- 从病毒扫描操作中排除文件, 71
- 共享文件, 46
- 启用病毒扫描, 69
- 扫描病毒, 69
- 添加病毒扫描引擎, 70
- 文件传输, 审计, 551–552

无

- 无效票证, 定义, 466

物

- 物理安全性, 说明, 36

系

- 系统, 防止危险程序, 122–123
- 系统 V IPC, 特权, 139
- 系统安全
 - root 访问限制, 62–63
 - su 命令监视, 61–62
 - UFS ACL, 115–116
 - 保存失败的登录尝试, 57
 - 登录访问限制, 36
 - 防止危险程序, 122–123
 - 更改
 - root 口令, 54
 - 基于角色的访问控制 (role-based access control, RBAC), 127–130
 - 任务列表, 122
 - 特权, 138–144
 - 显示
 - 没有口令的用户, 56
 - 用户的登录状态, 55

系统安全（续）

限制远程root 访问, 62–63

硬件保护, 63–65

系统安全性

root 访问限制, 47

su 命令监视, 42

登录访问限制, 36

防火墙, 49–50

访问, 35

概述, 35

基于角色的访问控制 (role-based access control, RBAC), 42

计算机访问, 36

口令, 36

口令加密, 37

受限 shell, 43, 44

特殊帐户, 39

硬件保护, 36

系统变量

另请参见变量

CRYPT_DEFAULT, 59

KEYBOARD_ABORT, 64–65

noexec_user_stack, 124

noexec_user_stack_log, 124

rstchown, 119

SYSLOG_FAILED_LOGINS, 58

系统调用

argument 审计标记, 563

exec_args 审计标记, 564

exec_env 审计标记, 564

ioctl 来清除音频设备, 90

return 审计标记, 568

系统属性, 相关特权, 139

系统信息库, 安装第三方提供者, 222

系统硬件, 控制访问, 63–65

显

显示

root 访问尝试, 62–63

su 命令尝试, 62–63

XML 格式的审计记录, 536

加密框架中的提供者, 218–222

可分配设备, 78–79

显示（续）

没有口令的用户, 56

您可以承担的角色, 149, 192

设备策略, 74

审计策略, 509

审计策略缺省值, 503–504

审计队列控制, 503–504, 511

审计记录, 535–537

审计记录定义, 530–532

审计缺省值, 503–504

文件和相关信息, 109

文件信息, 117–118

系统范围的审计的例外, 503–504

选定的审计记录, 532–533

用户的登录状态, 55

主体子列表 (Kerberos), 418

限

限制

超级用户, 61–63

审计文件大小, 549

用户特权, 160

远程超级用户访问, 62–63

在权限配置文件中使用权, 160

限制对 KDC 服务器的访问, 389–390

限制特权集, 141

向

向您的帐户授予访问权限, 453–455

消

消息验证代码 (message authentication code, MAC),
计算文件, 213–215

销

销毁, 使用 kdestroy 销毁票证, 450

效

效率, 审计和, 499

协

协调通用时间 (Coordinated Universal Time, UTC), 审计中使用的时戳, 532

写

写权限, 符号模式, 114

卸

卸载, 加密提供者, 224

新

新功能

安全 Shell 和 FIPS-140, 272

审计增强功能, 488-489

新增功能

SASL, 265

安全 Shell 增强功能, 271-272

星

星号 (*)

device_allocate 文件, 88, 89

检查 RBAC 授权, 162

通配符

在 RBAC 授权中, 188

修

修改

策略 (Kerberos), 435-436

角色 (RBAC), 167-168

用户 (RBAC), 168-169

修改 (续)

用户安全属性, 505-508

主体 (Kerberos), 424-425

主体的口令 (Kerberos), 425

选

选择

您的口令, 450-451

审计记录, 533-535

审计迹中的事件, 533-535

审计类, 504-505

压

压缩, 磁盘上的审计文件, 549-550

掩

掩码 (审计), 进程预选的说明, 559

验

验证

AUTH_DH 客户机-服务器会话, 248-251

DH 验证, 248-251

Kerberos 概述, 468

Kerberos 和, 301

安全 RPC, 247

安全 Shell

方法, 270-271

进程, 288

类型, 48-49

命名服务, 247

配置跨领域, 341-343

使用 -x 选项禁用, 457

说明, 48-49

网络安全性, 48-49

已挂载 NFS 的文件, 254

与 NFS 一起使用, 247

术语, 464-465

验证方法

- 安全 Shell, 270–271
- 安全 Shell 中的 GSS-API 凭证, 270
- 安全 Shell 中的公钥, 270
- 安全 Shell 中的口令, 270
- 安全 Shell 中基于主机, 270, 273–275

验证器, 返回到 NFS 客户机, 250

验证者, 在 Kerberos 中, 465

以

以后生效的票证

- 定义, 466
- 说明, 302

异

异步审计事件, 558–559

溢

溢出防止, 审计迹, 538–539

音

音频设备, 安全性, 90

应

应用服务器, 配置, 343–345

映

映射

- UID 到 Kerberos 主体, 473–474
- 事件到类 (审计), 481
- 主机名到领域 (Kerberos), 315

映射 GSS 凭证, 317

硬

硬件

- 保护, 36, 63–65
- 列出连接的硬件加速器, 227–228
- 要求提供访问口令, 64

硬件提供者

- 禁用加密机制, 228–230
- 列出, 227–228
- 启用机制和功能, 229
- 装入, 227

硬盘, 审计的空间要求, 499

永

永久审计策略, 配置的审计策略, 508–511

用

用户

- 安全 Shell 缺省值例外, 276–277
- 初始可继承特权, 141
- 创建
 - root 用户, 173–175
- 对角色进行验证, 172–173
- 对权限配置文件进行验证, 173
- 分配设备, 81–82
- 挂载分配的设备, 82–83
- 基本特权集, 141
- 计算文件的 MAC, 213–215
- 计算文件摘要, 212–213
- 加密文件, 215–218
- 将特权指定给, 169
- 解除分配设备, 84
- 禁止登录, 56–57
- 没有口令, 56
- 取消挂载分配的设备, 84
- 确定直接指定的特权, 176–177
- 确定自己的特权命令, 178–179
- 删除审计标志, 508
- 审计单个用户, 507–508
- 审计他们的所有命令, 544–545
- 生成对称密钥, 208–212
- 使用权限配置文件, 173

用户 (续)

- 为组创建权限配置文件, 508
- 显示登录状态, 55
- 限制基本特权, 160
- 修改审计预选掩码, 505-508
- 修改属性 (RBAC), 168-169
- 运行特权命令故障排除, 178-179
- 指定 RBAC 缺省值, 191
- 指定多个角色, 168-169
- 指定分配授权给, 78
- 指定权限配置文件, 169

用户 ID

- 审计 ID 和, 477-478, 559
- 在 NFS 服务中, 348

用户 ID 号 (UID), 特殊帐户和, 39**用户操作过程, 分配设备, 76-81****用户过程**

- chkey 命令, 253
- 保护文件, 117
- 承担角色, 149-150
- 创建自签名证书, 234-235
- 导出证书, 237-238
- 导入证书, 235-236
- 计算文件的 MAC, 213-215
- 计算文件摘要, 212-213
- 加密 NIS 用户的私钥, 253
- 加密文件, 205-206
- 将插件添加到 KMF 中, 243-244
- 解密文件, 215-218
- 生成对称密钥
 - 使用 dd 命令, 206-208
 - 使用 pktool 命令, 208-212
- 生成密钥库口令短语, 238
- 使用 pktool 命令, 233-234
- 使用安全 Shell, 277
- 使用指定的角色, 149-150

用户权限管理, 请参见特权**用户数据库 (RBAC), 请参见 user_attr 数据库****用户帐户****另请参见用户**

- 更改 root 口令, 54
- 显示登录状态, 55

用户主体, 说明, 306**邮**

- 邮件, 用于安全 Shell, 282

有

- 有效特权集, 141

预

- 预选, 审计类, 504-505
- 预选掩码 (审计), 说明, 559

远**远程登录**

- 安全性和, 250
- 防止超级用户, 62-63
- 授权, 48-49
- 验证, 48-49

允

- 允许特权集, 141

摘**摘要**

- 计算文件, 212-213
- 文件, 212-213

证**证书**

- 导出供其他系统使用, 237-238
- 导入密钥库, 235-236
- 签署 PKCS #10 CSR
 - 使用 pktool 命令, 242-243
- 通过 pktool gencert 命令生成, 234-235

证书签名请求 (certificate signing request, CSR), 请参见证书

执

执行权限, 符号模式, 114

直

直接领域, 342–343

指

指定

脚本中命令的特权, 183

角色的特权, 168

权限配置文件

到角色, 167–168

用户的特权, 169

在本地将角色指定给用户, 157–158

在权限配置文件中为命令指定特权, 161

终

终端 ID, 审计, 560

重

重定向箭头 (>), 防止重定向, 43

重放的事务, 250

重新启动

ssh 服务, 276

sshd 守护进程, 276

加密服务, 230

术

术语

Kerberos, 464–468

术语 (续)

特定于 Kerberos, 464

特定于验证的, 464–465

主

主, 在主体名称中, 305–306

主 KDC

从 KDC 服务器和, 307

定义, 464

交互配置, 324–325

自动配置, 323–324

主 KDC 服务器

从 KDC 服务器和, 322

交换从 KDC 服务器, 368–372

使用 LDAP 配置, 329–335

手动配置, 325–329

主机

安全 Shell 缺省值例外, 276–277

安全 Shell 主机, 270

禁用 Kerberos 服务, 445–446

可信主机, 50

主机名, 映射到领域, 315

主体

Kerberos, 305–306

SEAM Tool 面板, 437–440

查看列表, 417–419

查看属性, 419–421

查看主体子列表, 418

创建, 421–423

创建 clntconfig, 328, 334

创建 host, 328, 334

从密钥表删除服务主体, 443–444

从密钥表文件删除, 443

服务主体, 306

复制, 424

管理, 411–446

将服务主体添加至密钥表, 441, 442–443

删除, 425–426

设置缺省值, 426–427

添加管理, 327, 333

修改, 424–425

用户 ID 比较, 348

用户主体, 306

主体（续）

- 用于管理的任务列表, 416
- 主体名称, 305–306
- 自动创建, 416–417

注

- 注册提供者, 加密框架, 204

转

- 转换, 审计记录为可读格式, 536

传**传播**

- KDC 数据库, 317
- Kerberos 数据库, 372–373

资**资源控制**

- project.max-locked-memory, 140
- zone.max-locked-memory, 140
- 特权, 和, 140

自

- 自动创建主体, 416–417

自动登录

- 禁用, 456
- 启用, 456

自动配置

- Kerberos
 - 从 KDC 服务器, 335–336
 - 主 KDC 服务器, 323–324

组**组**

- 安全 Shell 缺省值例外, 276–277
- 更改文件所有权, 119

组件

- BART, 94–95
- RBAC, 130–132
- 安全 Shell 用户会话, 289
- 设备分配机制, 85–86

最

- 最小特权, 原则, 138
- 最小特权原则, 138

