在 Oracle® Solaris 11.2 中管理安全 Shell 访问

文件号码 E53966-02
2014 年 9 月

ORACLE

目录

使用本文档 5

1 使用安全 Shell（任务） 7
安全 Shell（概述） 7
安全 Shell 验证 8
安全 Shell 和 OpenSSH 项目 9
安全 Shell 和 FIPS 140 10
配置安全 Shell（任务） 11
配置安全 Shell（任务列表） 11
▼ 如何设置基于主机的安全 Shell 验证 12
▼ 如何在安全 Shell 中配置端口转发 14
▼ 如何创建安全 Shell 缺省设置的用户和主机例外 15
▼ 如何为 sftp 文件创建隔离的目录 16
使用安全 Shell（任务） 17
使用安全 Shell（任务列表） 17
▼ 如何生成用于安全 Shell 的公钥/私钥对 18
▼ 如何更改安全 Shell 私钥的口令短语 20
▼ 如何使用安全 Shell 登录到远程主机 20
▼ 如何减少安全 Shell 中的口令指示 22
▼ 如何使用安全 Shell 远程管理 ZFS 23
▼ 如何在安全 Shell 中使用端口转发 24
▼ 如何使用安全 Shell 复制文件 25
▼ 如何设置到防火墙外部的主机的缺省安全 Shell 连接 26

2 安全 Shell 参考 29
典型的安全 Shell 会话 29
安全 Shell 中的会话特征 29
安全 Shell 中的验证和密钥交换 30
安全 Shell 中的命令执行和数据转发 31
安全 Shell 中的客户机和服务器配置 31

安全 Shell 中的客户机配置 31
安全 Shell 中的服务器配置 32
安全 Shell 中的关键字 32
安全 Shell 中特定于主机的参数 35
安全 Shell 和登录环境变量 35
在安全 Shell 中维护已知主机 36
安全 Shell 文件 36
安全 Shell 命令 38

索引 41

使用本文档

《在 *Oracle*® *Solaris 11.2* 中管理安全 *Shell* 访问》介绍了如何管理和使用用于安全远程访问的安全 Shell 功能。

- **概述** – 介绍有关在 Oracle Solaris 中使用安全 Shell 的概念和任务。
- **目标读者** – 必须在企业中实现安全的系统管理员。
- **必备知识** – 熟悉安全概念和术语。

产品文档库

有关本产品的最新信息和已知问题均包含在文档库中，网址为：http://www.oracle.com/pls/topic/lookup?ctx=E56344。

获得 Oracle 支持

Oracle 客户可通过 My Oracle Support 获得电子支持。有关信息，请访问 http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info；如果您听力受损，请访问 http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs。

反馈

可以在 http://www.oracle.com/goto/docfeedback 上提供有关本文档的反馈。

♦ ♦ ♦ 第 1 章

1 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、远程窗口显示和远程文件传输的命令。本章包含以下主题：

- “安全 Shell（概述）” [7]
- “安全 Shell 和 OpenSSH 项目” [9]
- “安全 Shell 和 FIPS 140” [10]
- “配置安全 Shell（任务）” [11]
- “使用安全 Shell（任务）” [17]

有关参考信息，请参见第 2 章 安全 Shell 参考。

安全 Shell（概述）

安全 Shell 是新安装的 Oracle Solaris 系统上的缺省远程访问协议。Oracle Solaris 中的安全 Shell 是基于开源工具包 OpenSSL 构建的；该工具包可实现安全套接字层和传输层安全性。

Oracle Solaris 中提供了该工具包的两个不同版本。

- 版本 1.0.0 是用于运行安全 Shell 的缺省版本。
- 版本 0.9.8 实现了 FIPS-140FIPS 140（一个针对加密模块的美国政府计算机安全标准）。

  有关如何在 FIPS 140 模式下使用安全 Shell 的信息，请参见“安全 Shell 和 FIPS 140” [10]。

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虚拟专用网络 (Virtual Private Network, VPN)。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 支持安全 Shell 协议的第 2 版 (v2)。在客户机端，除了 v2，客户机还支持版本 1 (v1)。

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《面向开发者的 Oracle Solaris 11 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。下表显示了安全 Shell 验证方法和主机要求。

表 1-1 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“在安全 Shell 中获取 GSS 凭证” [30]。

基于主机	用户帐户
	
	`/etc/ssh/ssh_host_rsa_key` 或 `/etc/ssh/ssh_host_dsa_key` 中的本地主机私钥
	
	`/etc/ssh/ssh_config` 中的 `Hostbased Authentication yes`
		用户帐户
		
		`/etc/ssh/known_hosts` 或 `~/.ssh/known_hosts` 中的本地主机公钥
		
		`/etc/ssh/sshd_config` 中的 `Hostbased Authentication yes`
		
		`/etc/ssh/sshd_config` 中的 `IgnoreRhosts no`

验证方法	本地主机（客户机）要求	远程主机（服务器）要求
		/etc/ssh/shosts.equiv、/etc/hosts.equiv、~/.rhosts 或 ~/.shosts 中的本地主机项
基于口令	用户帐户	用户帐户 支持 PAM。
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
RSA 或 DSA 公钥	用户帐户 ~/.ssh/id_rsa 或 ~/.ssh/id_dsa 中的私钥 ~/.ssh/id_rsa.pub 或 ~/.ssh/id_dsa.pub 中的用户公钥	用户帐户 ~/.ssh/authorized_keys 中的用户公钥

安全 Shell 和 OpenSSH 项目

安全 Shell 是 OpenSSH (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。从 2012 年 9 月开始，Oracle Solaris 中的安全 Shell 版本为 2.0。ssh -V 命令显示版本号。

在此安全 Shell 发行版中，针对 v2 协议实现了以下功能：

- 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 – 对 ssh-keygen 命令生成的私钥使用 AES-128 算法进行保护。此算法可保护新生成的密钥和重新加密的密钥，如口令短语更改时。
- 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，请参见如何创建安全 Shell 缺省设置的用户和主机例外 [15]。

Oracle Solaris 工程师提供了对 OpenSSH 项目的错误修复。此外，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 OpenSSH 项目中的特权分离代码。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，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 Oracle Solaris 审计服务中。有关审计服务的详细信息，请参见《在 Oracle Solaris 11.2 中管理审计 》。
- 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 协议提供代理命令。有关示例，请参见如何设置到防火墙外部的主机的缺省安全 Shell 连接 [26]。

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。

安全 Shell 和 FIPS 140

安全 Shell 是 OpenSSL FIPS 140 模块的使用者。Oracle Solaris 为服务器端和客户端提供了 FIPS 140 选项。要符合 FIPS 140 要求，管理员应该配置和使用 FIPS 140 选项。

FIPS 模式（其中安全 Shell 使用 OpenSSL 的 FIPS 140 模式）不是缺省模式。作为管理员，您必须显式启用安全 Shell 以在 FIPS 140 模式下运行。可以使用命令 `ssh -o "UseFIPS140 yes"` *remote-host* 调用 FIPS 140 模式。此外，还可以在配置文件中设置关键字。

简而言之，此实现由以下项组成：

- 以下经 FIPS 140 认可的加密算法在服务器和客户端可用：`aes128-cbc`、`aes192-cbc` 和 `aes256-cbc`。

  `3des-cbc` 在客户端缺省可用，但由于潜在的安全风险，它不在服务器端的加密算法列表中。
- 以下经 FIPS 140 认可的消息验证代码 (Message Authentication Code, MAC) 可用：

  - hmac-sha1、hmac-sha1-96
  - hmac-sha2-256、hmac-sha2-256-96
  - hmac-sha2-512、hmac-sha2-512-96
- 支持四种服务器-客户机配置：
  - 客户端和服务器端均没有 FIPS 140 模式
  - 客户端和服务器端均有 FIPS 140 模式
  - 服务器端有 FIPS 140 模式，客户端没有
  - 客户端有 FIPS 140 模式，服务器端没有
- ssh-keygen 命令有一个选项用来生成在 FIPS 模式下安全 Shell 客户机需要的处于 PKCS #8 格式的用户私钥。有关更多信息，请参见 ssh-keygen(1) 手册页。

有关 FIPS 140 的更多信息，请参见《Using a FIPS 140 Enabled System in Oracle Solaris 11.2 》。另请参见 sshd(1M)、sshd_config(4)、ssh(1) 和 ssh_config(4) 手册页。

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。

配置安全 Shell（任务）

安全 Shell 是在安装时配置的。更改这些缺省设置需要管理干预。以下任务演示了如何更改某些缺省值。

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。要使用安全 Shell，请参见“使用安全 Shell（任务）” [17]。

任务	说明	有关说明
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。	如何设置基于主机的安全 Shell 验证 [12]
增大缓冲区大小以处理连接延迟。	对于高带宽高延迟网络，请增大 TCP 属性 recv_buf 的值。	《在 Oracle Solaris 11.2 中管理 TCP/IP 网络、IPMP 和 IP 隧道 》中的“更改 TCP 接收缓冲区大小”
配置端口转发。	允许用户使用端口转发。	如何在安全 Shell 中配置端口转发 [14]
配置安全 Shell 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安全 Shell 值。	如何创建安全 Shell 缺省设置的用户和主机例外 [15]

任务	说明	有关说明
隔离一个 root 环境以用于 sftp 传输。	为文件传输提供受保护的目录。	如何为 sftp 文件创建隔离的目录 [16]

▼ 如何设置基于主机的安全 Shell 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，“客户机”一词和“本地主机”一词是指用户在其中键入 ssh 命令的系统。术语服务器和远程主机是指客户机尝试访问的系统。

开始之前 您必须承担 root 角色。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

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，您或用户应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。

   - **如果是您进行配置，请将客户机作为一项添加到服务器的 /etc/ssh/shosts.equiv 文件中。**

     client-host

   - **如果是用户进行配置，则他们应将客户机的一个项添加到其在服务器上的 ~/.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. 使用以下方法之一，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，请参见如何生成用于安全 Shell 的公钥/私钥对 [18]。

   - 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"'
        ```

注 - 如果服务器中没有主机密钥，则使用安全 Shell 将生成类似如下的错误消息：

```
Client and server could not agree on a key exchange algorithm:
client "diffie-hellman-group-exchange-sha256,diffie-hellman-group-
exchange-sha1,diffie-hellman-group14-sha1,diffie-hellman-group1-sha1",
server "gss-group1-sha1-toWM5Slw5Ew8Mqkay+al2g==". Make sure host keys
are present and accessible by the server process. See sshd_config(4)
description of "HostKey" option.
```

---

     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>

例 1-1 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。例如，将出现针对用户 John Doe 的以下信息：

```
## /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 您必须承担 root 角色。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

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《在 Oracle Solaris 11.2 中管理系统服务 》中的第 1 章 “服务管理工具简介”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
   ```

▼ 如何创建安全 Shell 缺省设置的用户和主机例外

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 您必须是指定有 solaris.admin.edit/etc/ssh/sshd_config 授权的管理员。缺省情况下，root 角色拥有此授权。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. **打开 /etc/ssh/sshd_config 文件进行编辑。**

   ```
   # pfedit /etc/ssh/sshd_config
   ```

2. **配置用户、组、主机或地址以使用不同于缺省设置的安全 Shell 设置。**
   将 Match 块置于全局设置之后。

   注 - 文件的全局部分可能不会始终列出缺省设置。有关缺省设置，请参见 sshd_config(4) 手册页。

   例如，您可能有不允许使用 TCP 转发的用户。在以下示例中，public 组中的所有用户和以 test 开头的所有用户名都不能使用 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 块的语法的信息，请参见 sshd_config(4) 手册页。

▼ 如何为 sftp 文件创建隔离的目录

此过程配置为 sftp 传输专门创建的 sftponly 目录。用户无法看见传输目录外的任何文件或目录。

开始之前 您必须承担 root 角色。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. **在安全 Shell 服务器上，创建一个隔离的目录作为 chroot 环境。**

```
# groupadd sftp
# useradd -m -G sftp -s /bin/false sftponly
# chown root:root /export/home/sftponly
# mkdir /export/home/sftponly/WWW
# chown sftponly:staff /export/home/sftponly/WWW
```

在此配置中，/export/home/sftonly 是只有 root 帐户可以访问的 chroot 目录。用户对 sftponly/WWW 子目录具有写入权限。

2. **仍然是在该服务器上，配置 sftp 组的匹配块。**

在 /etc/ssh/sshd_config 文件中找到 sftp subsystem 项，并按如下方式修改此文件：

```
# pfedit /etc/ssh/sshd_config
...
# sftp subsystem
#Subsystem      sftp    /usr/lib/ssh/sftp-server
Subsystem       sftp    internal-sftp
...
```

```
## Match Group for Subsystem
## At end of file, to follow all global options
Match Group sftp
ChrootDirectory %h
ForceCommand internal-sftp
AllowTcpForwarding no
```

可以使用以下变量指定 chroot 路径：

- %h – 指定起始目录。
- %u – 指定经验证用户的用户名。
- %% – 对 % 符号进行转义。

3. **在客户机上，验证配置可正常运行。**
   您的 chroot 环境中的文件可能有所不同。

```
root@client:~# ssh sftponly@server
This service allows sftp connections only.
Connection to server closed.      No shell access, sftp is enforced.
root@client:~# sftp sftponly@server
sftp> pwd      sftp access granted
Remote working directory: /     chroot directory looks like root directory
sftp> ls
WWW             local.cshrc     local.login     local.profile
sftp> get local.cshrc
Fetching /local.cshrc to local.cshrc
/local.cshrc    100%  166     0.2KB/s   00:00     user can read contents
sftp> put /etc/motd
Uploading /etc/motd to /motd
Couldn't get handle: Permission denied     user cannot write to / directory
sftp> cd WWW
sftp> put /etc/motd
Uploading /etc/motd to /WWW/motd
/etc/motd     100%  118     0.1KB/s   00:00     user can write to WWW directory
sftp> ls -l
-rw-r--r--    1 101  10    118 Jul 20 09:07 motd     successful transfer
sftp>
```

使用安全 Shell（任务）

本部分提供让用户熟悉安全 Shell 的过程。

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	有关说明
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。	如何生成用于安全 Shell 的公钥/私钥对 [18]
更改口令短语。	更改用于验证私钥的短语。	如何更改安全 Shell 私钥的口令短语 [20]
使用安全 Shell 登录。	远程登录时，提供加密的安全 Shell 通信。	如何使用安全 Shell 登录到远程主机 [20]
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。	如何减少安全 Shell 中的口令指示 [22]
以 root 用户身份登录到安全 Shell。	允许作为 root 登录以使用 ZFS send 和 receive 命令。	如何使用安全 Shell 远程管理 ZFS [23]
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。	如何在安全 Shell 中使用端口转发 [24]
使用安全 Shell 复制文件。	在主机之间安全地复制文件。	如何使用安全 Shell 复制文件 [25]
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。	如何设置到防火墙外部的主机的缺省安全 Shell 连接 [26]

▼ 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 向系统管理员询问是否配置了基于主机的验证。

1. **启动密钥生成程序。**

```
mySystem% 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。

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。可通过按回车键或提供替代文件名来选择此文件。

```
Enter file in which to save the key (/home/username/.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/username/.ssh/id_rsa.
Your public key has been saved in /home/username/.ssh/id_rsa.pub.
The key fingerprint is:
0e:fb:3d:57:71:73:bf:58:b8:eb:f3:a3:aa:df:e0:d1 username@my

System
```

5. 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. 根据您的网络的验证方法，通过使用相应的选项来登录远程主机。

- 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，请参见如何使用安全 Shell 登录到远程主机 [20]。

  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）。

  ```
  mySystem% cat $HOME/.ssh/id_rsa.pub | ssh myRemoteHost \
  'cat >> .ssh/authorized_keys && echo "Key copied"'
  ```

  复制该文件后，将显示 "Key copied"（密钥已复制）消息。

  b. 出现提示时，提供登录口令。

```
Enter password:       Type login password
Key copied
mySystem%
```

7. （可选） 避免在将来提示输入口令短语。

   请参见如何减少安全 Shell 中的口令指示 [22]。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

以下命令更改私钥的验证机制（即口令短语），而不更改实际的私钥。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。

```
mySystem% ssh-keygen -p
Enter file which contains the private key
      (/home/username/.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。

```
mySystem% ssh myRemoteHost -l username
```

2. 如果出现提示，请检验远程主机密钥的真实性。

   此时可能会出现提示，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。

   - 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/username/.ssh/id_rsa':      <Type passphrase>
```

   b. 出现提示时，键入帐户口令。

```
username@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
mySystem%
```

例 1-2 在安全 Shell 中显示远程 GUI

在此示例中，jdoe 在两个系统上都是初始用户，并且指定有 Software Installation 权限配置文件。jdoe 想要使用远程系统上的软件包管理器 GUI。X11Forwarding 关键字的缺省值仍然是 yes，并且 xauth 软件包安装在远程系统上。

```
% ssh -l jdoe -X myRemoteHost
jdoe@myRemoteHost's password: password
Last login: Wed Sep  7 09:07:49 2011 from myLocalHost
Oracle Corporation      SunOS 5.11       September 2011
myRemoteHost% packagemanager &
```

▼ 如何减少安全 Shell 中的口令指示

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。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。

```
mySystem% eval `ssh-agent`
Agent pid 9892
```

2. 检验是否已启动代理守护进程。

```
mySystem% pgrep ssh-agent
9892
```

3. 将私钥添加到代理守护进程。

```
mySystem% ssh-add
Enter passphrase for /home/username/.ssh/id_rsa:      <Type passphrase>
Identity added: /home/username/.ssh/id_rsa(/home/username/.ssh/id_rsa)
mySystem%
```

4. 启动安全 Shell 会话。

```
mySystem% ssh myRemoteHost -l username
```

系统不会提示您输入口令短语。

例 1-3 使用 ssh-add 选项

在本示例中，jdoe 将向代理守护进程添加两个密钥。-l 选项用于列出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
mySystem% ssh-add
Enter passphrase for /home/jdoe/.ssh/id_rsa:      <Type passphrase>
Identity added: /home/jdoe/.ssh/id_rsa(/home/jdoe/.ssh/id_rsa)
mySystem% 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)

mySystem% 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)
```

▼ 如何使用安全 Shell 远程管理 ZFS

缺省情况下，root 角色无法使用安全 Shell 进行远程登录。过去，root 曾使用安全 Shell 执行许多重要任务，例如将 ZFS 池数据发送到远程系统上的存储。在此过程中，root 角色创建一个可担任远程 ZFS 管理员的用户。

开始之前 您必须承担 root 角色。有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全 》中的“使用所指定的管理权限”。

1. **在两个系统上都创建该用户。**

   例如，创建 zfsroot 用户，并提供口令。

```
source # useradd -c "Remote ZFS Administrator" -u 1201 -d /home/zfsroot zfsroot
source # passwd zfsroot
Enter password:
Retype password:
#

dest # useradd -c "Remote ZFS Administrator" -u 1201 -d /home/zfsroot zfsroot
dest # passwd zfsroot
...
```

   必须在两个系统上以完全相同的方式定义 zfsroot 用户。

2. **创建用于安全 Shell 验证的用户密钥对。**

   密钥对是在源系统上创建的。然后，将公钥复制到目标系统上的 zfsroot 用户。

   a. **生成密钥对并将其放置在文件 id_migrate 中。**

```
# ssh-keygen -t rsa -P "" -f ~/id_migrate
Generating public/private rsa key pair.
Your identification has been saved in /root/id_migrate.
Your public key has been saved in /root/id_migrate.pub.
The key fingerprint is:
3c:7f:40:ef:ec:63:95:b9:23:a2:72:d5:ea:d1:61:f0 root@source
```

   b. **将密钥对的公用部分发送到目标系统。**

```
# scp ~/id_migrate.pub zfsroot@dest:
The authenticity of host 'dest (10.134.76.126)' can't be established.
RSA key fingerprint is 44:37:ab:4e:b7:2f:2f:b8:5f:98:9d:e9:ed:6d:46:80.
Are you sure you want to continue connecting (yes/no)? yes
```

```
Warning: Permanently added 'dest,10.134.76.126' (RSA) to the list of known hosts.
Password:
id_migrate.pub 100% |*****************************| 399 00:00
```

3. 在两个系统上，将 ZFS File Management 权限配置文件指定给 **zfsroot**。

```
source # usermod -P +'ZFS File System Management' -S files zfsroot
dest # usermod -P +'ZFS File System Management' -S files zfsroot
```

4. 验证是否已为目标系统指定了该权限配置文件。

```
dest # profiles zfsroot
zfsroot:
ZFS File System Management
Basic Solaris User
All
```

5. 在目标系统上，将密钥对的公用部分移动到私有的 **/home/zfsroot/.ssh** 目录。

```
root@dest # su - zfsroot
Oracle Corporation      SunOS 5.11      11.1    May 2012
zfsroot@dest $ mkdir -m 700 .ssh
zfsroot@dest $ cat id_migrate.pub >> .ssh/authorized_keys
```

6. 验证配置是否正常工作。

```
root@source# ssh -l zfsroot -i ~/id_migrate dest \
pfexec /usr/sbin/zfs snapshot zones@test
root@source# ssh -l zfsroot -i ~/id_migrate dest \
pfexec /usr/sbin/zfs destroy zones@test
```

7. （可选） 验证您可以创建快照并复制数据。

```
root@source# zfs snapshot -r rpool/zones@migrate-all
root@source#  zfs send -rc rpool/zones@migrate-all | \
ssh -l zfsroot -i ~/id_migrate dest pfexec /usr/sbin/zfs recv -F zones
```

8. （可选） 删除使用 **zfsroot** 帐户进行 ZFS 管理的功能。

```
root@dest# usermod -P -'ZFS File System Management' zfsroot
root@dest#  su - zfsroot
zfsroot@dest#  cp .ssh/authorized_keys .ssh/authorized_keys.bak
zfsroot@dest#  grep -v root@source .ssh/authorized_keys.bak> .ssh/authorized_keys
```

▼ 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，管理员必须先在远程安全 Shell 服务器上启用端口转发。有关详细信息，请参见如何在安全 Shell 中配置端口转发 [14]。

- **设置从远程端口到本地端口或者从本地端口到远程端口的安全端口转发。**

  - **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。

    ```
    mySystem% 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。

    ```
    mySystem% ssh -R remotePort:localhost:localPort
    ```

例 1-4 使用本地端口转发接收邮件

以下示例说明如何使用本地端口转发来安全地接收来自远程服务器的邮件。

```
myLocalHost% ssh -L 9143:myRemoteHost:143 myRemoteHost
```

此命令可将连接从 myLocalHost 上端口 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。

例 1-5 使用远程端口转发在防火墙外部进行通信

本示例说明企业环境中的用户如何将连接从外部网络中的主机转发到公司防火墙内的主机。

```
myLocalHost% ssh -R 9022:myLocalHost:22myOutsideHost
```

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《在 Oracle Solaris 11.2 中管理远程系统 》中的“使用 scp 命令进行远程复制”和 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，请参见例 1-6 “使用 sftp 命令时指定一个端口” 和《在 Oracle Solaris 11.2 中管理远程系统 》中的“登录到远程系统以复制文件 (sftp)”。

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。有关更多信息，请参见《在 Oracle Solaris 11.2 中管理审计 》中的“如何审计 FTP 和 SFTP 文件传输”。

---

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。

   ```
   mySystem% scp myfile.1 username@myRemoteHost:~
   ```

2. **出现提示时，提供口令短语。**

   ```
   Enter passphrase for key '/home/username/.ssh/id_rsa':      <Type passphrase>
   myfile.1       25% |*******                     |    640 KB  0:20 ETA
   myfile.1
   ```

   在键入口令短语后，会显示一个进度条，如输出中的第二行中所示。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）

例 1-6 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的主机的缺省安全 Shell 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，请参见例 1-7 “通过安全 Shell 命令行连接到防火墙外部的主机”。

您可以通过自己的个人配置文件 `~/.ssh/config` 定制您的 `ssh` 交互，也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：

   ```
   mySystem% 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。

例 1-7 通过安全 Shell 命令行连接到防火墙外部的主机

如何设置到防火墙外部的主机的缺省安全 Shell 连接 [26]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

♦ ♦ ♦ 第 2 章

2 安全 Shell 参考

本章介绍了 Oracle Solaris 的安全 Shell 功能中的配置选项，并涵盖了以下主题：

- “典型的安全 Shell 会话” [29]
- “安全 Shell 中的客户机和服务器配置” [31]
- “安全 Shell 中的关键字” [32]
- “在安全 Shell 中维护已知主机” [36]
- “安全 Shell 文件” [36]
- “安全 Shell 命令” [38]

有关配置安全 Shell 的过程，请参见第 1 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` 设置。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 1-1 “安全 Shell 的验证方法” 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，这通常需要使用口令。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。

有关在安全 RPC 中使用 `mech_dh` 的更多信息，请参见《在 Oracle Solaris 11.2 中管理 Kerberos 和其他验证服务 》中的第 10 章 “配置网络服务验证”。有关使用 `mech_krb5` 的更多信息，请参见《在 Oracle Solaris 11.2 中管理 Kerberos 和其他验证

服务 》中的第 2 章 “关于 Kerberos 服务”。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 会话的特征通过配置文件控制。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机请求。有关服务器配置设置的列表，请参见“安全 Shell 中的关键字” [32]。有关详细信息，请参见 sshd_config(4) 手册页。

“安全 Shell 中的关键字” [32]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。这些关键字在 ssh(1)、scp(1)、sftp(1) 和 ssh_config(4) 手册页中有详细介绍。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 2-5 “安全 Shell 关键字的命令行等效项”。

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。“安全 Shell 中的关键字” [32]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 2-1 安全 Shell 配置文件中的关键字

关键字	缺省值	位置
AllowGroups		服务器
AllowTcpForwarding	yes	服务器
AllowUsers		服务器
AuthorizedKeysFile	~/.ssh/authorized_keys	服务器
Banner	/etc/issue	服务器
Batchmode	no	客户机
BindAddress		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		客户机
ConnectionAttempts	1	客户机
ConnectTimeout	系统 TCP 超时	客户机
DenyGroups		服务器
DenyUsers		服务器
DisableBanner	no	客户机
DynamicForward		客户机

关键字	缺省值	位置
EscapeChar	~	客户机
FallBackToRsh	no	客户机
ForwardAgent	no	客户机
ForwardX11	no	客户机
ForwardX11Trusted	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“安全 Shell 中特定于主机的参数” [35]。	客户机
HostbasedAuthentication	no	两者
HostbasedUsesNameFromPacketOnly	no	服务器
HostKey (v1)	/etc/ssh/ssh_host_key	服务器
HostKey (v2)	/etc/ssh/host_rsa_key、 /etc/ssh/host_dsa_key	服务器
HostKeyAlgorithms	ssh-rsa、ssh-dss	客户机
HostKeyAlias		客户机
HostName		客户机
IdentityFile	~/.ssh/id_dsa、~/.ssh/id_rsa	客户机
IgnoreIfUnknown		客户机
IgnoreRhosts	yes	服务器
IgnoreUserKnownHosts	yes	服务器
KbdInteractiveAuthentication	yes	两者
KeepAlive	yes	两者
KeyRegenerationInterval	3600（秒）	服务器
ListenAddress		服务器
LocalForward		客户机
LoginGraceTime	120（秒）	服务器
LogLevel	info	两者
LookupClientHostnames	yes	服务器
MACs	hmac-sha1-*、hmac-md5-* 和 hmac-sha2-* 算法。	两者
Match		服务器
MaxStartups	10:30:60	服务器
NoHostAuthenticationForLocalHost	no	客户机

关键字	缺省值	位置
NumberOfPasswordPrompts	3	客户机
PAMServiceName		服务器
PAMServicePrefix		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、 passwor	客户机
PreUserauthHook		服务器
PrintLastLog	yes	服务器
PrintMotd	no	服务器
Protocol	2,1	两者
ProxyCommand		客户机
PubkeyAuthentication	yes	两者
RekeyLimit	1G 至 4G	客户机
RemoteForward		客户机
RhostsAuthentication	no	服务器, v1
RhostsRSAAuthentication	no	服务器, v1
RSAAuthentication	no	服务器, v1
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
UseFIPS140	no	两者
UseOpenSSLEngine	yes	两者
UsePrivilegedPort	no	两者
User		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

关键字	缺省值	位置
X11UseLocalHost	yes	服务器
XAuthLocation	/usr/bin/xauth	两者

安全 Shell 中特定于主机的参数

有时，不同的本地主机有不同的安全 Shell 特征很有用。管理员可以对 /etc/ssh/ssh_config 文件中的条目按 Host 关键字分组，从而在该文件中定义不同的、要根据主机或正则表达式进行应用的参数集。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。

/etc/default/login 中的项	**sshd_config** 中的关键字和值
CONSOLE=*	PermitRootLogin=without-password
#CONSOLE=*	PermitRootLogin=yes
PASSREQ=YES	PermitEmptyPasswords=no
PASSREQ=NO	PermitEmptyPasswords=yes
#PASSREQ	PermitEmptyPasswords=no
TIMEOUT=*seconds*	LoginGraceTime=*seconds*
#TIMEOUT	LoginGraceTime=120
RETRIES 和 SYSLOG_FAILED_LOGINS	仅适用于 password 和 keyboard-interactive 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，sshd 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 ALTSHELL=YES，其中 sshd 守护进程使用用户的 shell 值。当 ALTSHELL=NO 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 /usr/bin。

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 文件只能按如下方式通过安全机制分发：

- 通过安全连接，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，可以使用采用了 scp 命令的脚本来复制最新版本。

安全 Shell 文件

下表列出了主要的安全 Shell 文件和建议的文件权限。

表 2-2　安全 Shell 文件

文件名	说明	建议的权限和所有者
~/.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*
~/.ssh/authorized_keys	包含获许以用户帐户登录的用户的公钥。	-rw-r--r-- *username*
~/.ssh/config	配置用户设置，该设置将覆盖系统设置。	-rw-r--r-- *username*
~/.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*
/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
~/.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*

文件名	说明	建议的权限和所有者
/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
~/.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/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
/etc/ssh/ssh_config	配置客户机系统上的系统设置。	-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
/etc/ssh_host_key.pub 或 /etc/ssh/ssh_host_dsa_key.pub 或 /etc/ssh/ssh_host_rsa_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
/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
/etc/ssh/sshd_config	包含安全 Shell 守护进程 sshd 的配置数据。	-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
/etc/ssh/sshrc	包含管理员指定的主机特定的初始化例程。	-rw-r--r-- root

注 - 可以用来自站点定制软件包的文件覆盖 sshd_config 文件。有关更多信息，请参见 pkg(5) 手册页中 overlay 文件属性的定义。

下表列出了可被关键字或命令选项覆盖的安全 Shell 文件。

表 2-3 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 /etc/ssh/host_dsa_key	HostKey	
~/.ssh/identity ~/.ssh/id_dsa, ~/.ssh/id_rsa	IdentityFile	ssh -i *ID-file* scp -i *ID-file*
~/.ssh/authorized_keys	AuthorizedKeysFile	
/etc/ssh/ssh_known_hosts	GlobalKnownHostsFile	
~/.ssh/known_hosts	UserKnownHostsFile	

文件名	关键字覆盖	命令行覆盖
	IgnoreUserKnownHosts	

安全 Shell 命令

下表汇总了主要的安全 Shell 命令。

表 2-4 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。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 2-5 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 *bind-addr*	scp -a *bind-addr*
Cipher	ssh -c *cipher*	scp -c *cipher*
Ciphers	ssh -c *cipher-spec*	scp -c *cipher-spec*
Compression	ssh -C	scp -C
DynamicForward	ssh -D *SOCKS4-port*	
EscapeChar	ssh -e *escape-char*	

关键字	**ssh** 命令行覆盖	**scp** 命令行覆盖
ForwardAgent	ssh -A（启用） ssh -a（禁用）	
ForwardX11	ssh -X（启用） ssh -x（禁用）	
GatewayPorts	ssh -g	
IPv4	ssh -4	scp -4
IPv6	ssh -6	scp -6
LocalForward	ssh -L *localport:remotehost:remoteport*	
MACS	ssh -m *MAC-spec*	
Port	ssh -p *port*	scp -P *port*
Protocol	ssh -2（仅限 v2）	
RemoteForward	ssh -R *remoteport:localhost:localport*	

索引

数字和符号

.rhosts 文件
  说明，36
.shosts 文件
  说明，36
/etc/default/login 文件
  安全 Shell 和，35
  说明，37
/etc/hosts.equiv 文件
  说明，36
/etc/nologin 文件
  说明，37
/etc/ssh_host_dsa_key.pub 文件
  说明，37
/etc/ssh_host_key.pub 文件
  说明，37
/etc/ssh_host_rsa_key.pub 文件
  说明，37
/etc/ssh/shosts.equiv 文件
  说明，37
/etc/ssh/ssh_config 文件
  主机特定的参数，35
  关键字，32
  覆盖，37
  说明，37
  配置安全 Shell，31
/etc/ssh/ssh_host_dsa_key 文件
  说明，37
/etc/ssh/ssh_host_key 文件
  覆盖，37
/etc/ssh/ssh_host_rsa_key 文件
  说明，37
/etc/ssh/ssh_known_hosts 文件
  安全分发，36
  控制分发，36
  覆盖，37
  说明，37
/etc/ssh/sshd_config 文件
  关键字，32
  说明，37
/etc/ssh/sshrc 文件
  说明，37
/system/volatile/sshd.pid 文件
  说明，37
~/.rhosts 文件
  说明，36
~/.shosts 文件
  说明，36
~/.ssh/authorized_keys 文件
  覆盖，37
  说明，36
~/.ssh/config 文件
  覆盖，37
  说明，36
~/.ssh/environment 文件
  说明，36
~/.ssh/id_dsa 文件
  覆盖，37
~/.ssh/id_rsa 文件
  覆盖，37
~/.ssh/identity 文件
  覆盖，37
~/.ssh/known_hosts 文件
  覆盖，37
  说明，36
~/.ssh/rc 文件
  说明，37
3des 加密算法
  ssh_config 文件，32
3des-cbc 加密算法
  ssh_config 文件，32

A
安全 Shell
  FIPS 140 支持，10
  OpenSSH 中的基础，9
  scp 命令，25
  TCP 和，14
  xauth 软件包，21
  使用端口转发，24
  公钥验证，8
  关键字，32
  典型会话，29
  减少登录时的提示，22
  创建密钥，18
  协议版本，8
  命令执行，31
  命名标识文件，36
  在没有口令时使用，22
  复制文件，25
  当前发行版中的更改，9
  指定系统缺省设置的例外，15
  数据转发，31
  文件，36
  更改口令短语，20
  本地端口转发，25，25
  生成密钥，18
  用户过程，17
  登录以显示远程 GUI，21
  登录环境变量，35
  登录远程主机，20
  管理，29
  管理 ZFS，23
  管理员任务列表，11
  说明，7
  跨防火墙连接，26
  转发邮件，25
  远程端口转发，25
  连接防火墙外部
    从命令行，28
    从配置文件，26
  配置 chroot 目录，16
  配置客户机，31
  配置服务器，32
  配置端口转发，14
  验证
    要求，8
  验证方法，8
    验证步骤，30
安全 Shell 中的 ALTSHELL，35
安全 Shell 中的 CONSOLE，35
安全 Shell 中的 PASSREQ，35
安全 Shell 中的 PATH，35
安全 Shell 中的 RETRIES，35
安全 Shell 中的 SUPATH，36
安全 Shell 中的 TIMEOUT，35
安全 Shell 中的 TZ，35
安全 Shell 中的端口转发，14，25
安全 Shell 中的验证
  方法，8
  进程，30
安全连接
  登录，20
  跨防火墙，26
安全性
  安全 Shell，7
  跨不安全的网络，26
aes128-cbc 加密算法
  ssh_config 文件，32
aes128-ctr 加密算法
  ssh_config 文件，32
AllowTcpForwarding 关键字
  更改，15
arcfour 加密算法
  ssh_config 文件，32
authorized_keys 文件
  说明，36

B
保护
  sftp 传输目录，16
变量
  login 和安全 Shell，35
  代理服务器和端口，27
  安全 Shell 中的设置，35
Blowfish 加密算法
  ssh_config 文件，32
blowfish-cbc 加密算法
  ssh_config 文件，32

C
创建
  安全 Shell 密钥，18
重新启动
  ssh 服务，15
  sshd 守护进程，15
chroot 目录
  sftp 和，16

D
代理守护进程
  安全 Shell，22
登录
  使用安全 Shell，20，20
  使用安全 Shell 显示 GUI，21
default/login 文件
  说明，37

F
防火墙系统
  使用安全 Shell 的外部连接
    从命令行，28
    从配置文件，26
  安全主机连接，26
访问
  使用安全 Shell 进行登录验证，22
  安全性
    登录验证，22
    远程系统，7
服务器
  为安全 Shell 配置，32
复制
  使用安全 Shell 复制文件，25
FIPS 140 支持
  使用 Sun Crypto Accelerator 6000 卡的安全 Shell，10
  安全 Shell 远程访问，10

G
更改
  安全 Shell 的口令短语，20
公钥
  安全 Shell 中的验证，8
  安全 Shell 身份文件，36
  更改口令短语，20
  生成公钥/私钥对，18
关键字，29
  参见 特定关键字
  安全 Shell，32
  安全 Shell 中的命令行覆盖，38
管理
  使用安全 Shell 进行远程登录，18
  使用安全 Shell 远程管理 ZFS，23
管理安全 Shell
  任务列表，11
  客户机，31
  服务器，32
  概述，29
GSS-API
  安全 Shell 中的凭证，30
  安全 Shell 中的验证，8

H
环境变量
  与 ssh-agent 命令一起使用，38
  安全 Shell 和，35
  覆盖代理服务器和端口，27
hmac-sha2 加密算法
  ssh_config 文件，33
  sshd_config 文件，33
Host 关键字
  ssh_config 文件，35
hosts.equiv 文件
  说明，36

I
IP 地址
  安全 Shell 检查，32
  安全 Shell 缺省设置的例外，15

J
基于主机的验证
  在安全 Shell 中配置，12
  说明，8

加密
  主机间的网络通信，7
  主机间通信，21
  在 ssh_config 文件中指定算法，32

K
客户机
  为安全 Shell 配置，29，31
口令
  在安全 Shell 中消除，22
  安全 Shell 中的验证，8
口令短语
  在安全 Shell 中使用，22
  安全 Shell 更改，20
  示例，21
known_hosts 文件
  控制分发，36
  说明，36

L
-l 选项
  ssh 命令，20
-L 选项
  ssh 命令，24
login 环境变量
  安全 Shell 和，35

M
密钥
  为安全 Shell 生成，18
命令
  安全 Shell 命令，38
命令执行
  安全 Shell，31
命名约定
  安全 Shell 身份文件，36
Match 块
  chroot 目录和，16
  安全 Shell 缺省设置的例外，15
mech_dh 机制
  GSS-API 凭证，30
mech_krb 机制
  GSS-API 凭证，30

N
nologin 文件
  说明，37

O
OpenSSH 项目，9 见 安全 Shell

P
配置
  基于主机的安全 Shell 验证，12
  安全 Shell
    客户机，31
    服务器，32
  安全 Shell 中的端口转发，14
  安全 Shell 任务列表，11
  安全 Shell 系统缺省设置的例外，15
  用于 sftp 的 chroot 目录，16
配置文件
  安全 Shell，29

R
任务列表
  使用安全 Shell，17
  配置安全 Shell，11
-R 选项
  ssh 命令，24

S
身份文件（安全 Shell）
  命名约定，36
生成用于安全 Shell 的密钥，18
使用安全 Shell, 任务列表，17
手册页
  安全 Shell，38
守护进程
  ssh-agent，22
  sshd，29
数据转发

安全 Shell，31
私钥
安全 Shell 身份文件，36
算法
ssh-keygen 中的口令短语保护，9
scp 命令
复制文件，25
说明，38
sftp 命令
chroot 目录和，16
复制文件，26
说明，38
shosts.equiv 文件
说明，37
SMF
ssh 服务，15
重新启动安全 Shell，15
ssh_config 文件
主机特定的参数，35
关键字，32 见 特定关键字
覆盖，37
配置安全 Shell，31
ssh_host_dsa_key 文件
说明，37
ssh_host_dsa_key.pub 文件
说明，37
ssh_host_key 文件
覆盖，37
ssh_host_key.pub 文件
说明，37
ssh_host_rsa_key 文件
说明，37
ssh_host_rsa_key.pub 文件
说明，37
ssh_known_hosts 文件，37
ssh 命令
使用，20
使用代理命令，28
端口转发选项，24
覆盖关键字设置，38
说明，38
远程管理 ZFS，23
ssh-add 命令
存储私钥，22
示例，22，22
说明，38
ssh-agent 命令
从命令行，22
说明，38
ssh-agent 守护进程，22
ssh-keygen 命令
使用，18
口令短语保护，9
说明，38
ssh-keyscan 命令
说明，38
ssh-keysign 命令
说明，38
.ssh/config 文件
覆盖，37
说明，36
.ssh/environment 文件
说明，36
.ssh/id_dsa 文件，37
.ssh/id_rsa 文件，37
.ssh/identity 文件，37
.ssh/known_hosts 文件
覆盖，37
说明，36
.ssh/rc 文件
说明，37
sshd 命令
说明，38
sshd_config 文件
/etc/default/login 项的覆盖，35
关键字，32 见 特定关键字
说明，37
sshd.pid 文件
说明，37
sshrc 文件
说明，37
Sun Crypto Accelerator 6000 板
安全 Shell 和 FIPS 140，10
SunSSH 见 安全 Shell
svcadm 命令, 重新启动安全 Shell，15
SYSLOG_FAILED_LOGINS
在安全 Shell 中，35

T
通配符
安全 Shell 中的主机，27
TCP, 安全 Shell 和，14，31

U
UDP
安全 Shell 和，14
端口转发和，14

V
v1 协议
安全 Shell，8
v2 协议
安全 Shell，8

W
伪 TTY
在安全 Shell 中使用，31
文件
使用安全 Shell 复制，25
用于管理安全 Shell，36

X
新增功能
安全 Shell 和 FIPS 140，10
安全 Shell 增强功能，9
-X 选项
ssh 命令，21
X11 转发
在 ssh_config 文件中配置，33，33
在安全 Shell 中，31
xauth 命令
X11 转发，35

Y
验证方法
安全 Shell，8
安全 Shell 中基于主机，8，12
安全 Shell 中的 GSS-API 凭证，8
安全 Shell 中的公钥，9
安全 Shell 中的口令，9
用户
安全 Shell 缺省设置的例外，15
用户过程
使用安全 Shell，17
邮件
使用安全 Shell，25

Z
主机
安全 Shell 主机，8
安全 Shell 缺省设置的例外，15
组
安全 Shell 缺省设置的例外，15
组件
安全 Shell 用户会话，31