

Secure Shell

Installation and Administration

May 2002

Product Version: Secure Shell, Version 1.1

Operating System and Version: Tru64 UNIX Version 5.1A or higher

This manual describes how to install, configure, and use the Secure Shell software.

© 2002 Hewlett-Packard Company

Compaq, the Compaq logo, Tru64, and TruCluster, are trademarks of Compaq Information Technologies Group, L.P. in the U.S. and/or other countries.

Microsoft and Windows NT are registered trademarks of Microsoft Corporation in the U.S. and/or other countries. UNIX is a trademark of The Open Group in the U.S. and/or other countries.

Contains SSH Secure Shell technology, SSH is a registered trademark of SSH Communications Security Corp. (<http://www.ssh.com>)

All other product names mentioned herein may be the trademarks of their respective companies.

Confidential computer software. Valid license from Compaq Computer Corporation, a wholly owned subsidiary of Hewlett-Packard Company, required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

None of Compaq, HP, or any of their subsidiaries shall be liable for technical or editorial errors or omissions contained herein. The information is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for HP or Compaq products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.

Contents

About This Manual

1 Secure Shell Overview

1.1	The Secure Shell Server	1-2
1.2	The Secure Shell Client	1-2
1.3	Server and Client Communication	1-2

2 Installing the Secure Shell Software

2.1	Downloading the Secure Shell Software	2-1
2.2	Upgrading the Secure Shell Software	2-2
2.3	Installing the Secure Shell Software	2-2
2.4	Installing and Viewing the Secure Shell Documentation	2-2
2.5	Starting the Server Software	2-3
2.6	Secure Shell Directories and Files	2-3
2.7	Deinstalling the Secure Shell Software	2-3
2.8	Removing the Secure Shell Documentation	2-4

3 Configuring and Managing the Secure Shell Software

3.1	Configuring the Server	3-1
3.2	Configuring the Client	3-4
3.2.1	Configuring Network Commands to Use Secure Shell	3-7
3.3	Configuring User Authentication	3-7
3.3.1	Configuring Password Authentication	3-8
3.3.2	Configuring Public Key Authentication	3-9
3.3.2.1	Configuring Public Key Authentication on the Client	3-9
3.3.2.2	Configuring Public Key Authentication on the Server	3-12
3.3.2.3	Accessing a Remote Server	3-13
3.3.2.4	Restricting User Access	3-13
3.3.2.5	Managing Passphrases	3-13
3.3.3	Configuring Host-Based Authentication	3-14
3.4	Managing the Server	3-16
3.4.1	Starting, Stopping, Restarting, and Resetting the sshd2 Daemon	3-16
3.4.2	Restricting Users to Home Directories	3-17

3.4.3	Creating a Public and Private Host Key	3-18
3.4.4	Forwarding TCP/IP and X11 Data Through a Secure Shell Connection	3-18
3.4.4.1	TCP/IP Port Forwarding	3-18
3.4.4.2	X11 Forwarding	3-19

4 Using the Secure Shell Commands

4.1	Copying Files	4-1
4.1.1	Using the scp2 Command	4-1
4.1.2	Using the sftp2 Command	4-2
4.2	Logging In and Executing Commands on a Server	4-2

Index

Examples

3-1	Sample sshd2_config File	3-2
3-2	Sample ssh2_config File	3-4
3-3	Public Key Authentication Login Output	3-13

Tables

1-1	Secure Shell Commands	1-1
-----	-----------------------------	-----

About This Manual

This manual describes how to install, configure, and use the Secure Shell software on a system running the HP Tru64 UNIX operating system software.

Audience

This manual is intended for anyone who is responsible for installing and configuring the Secure Shell sever and client software on a system running the Tru64 UNIX operating system software.

Organization

The manual is organized as follows:

- Chapter 1* Provides an overview of Secure Shell client and server software.
- Chapter 2* Describes how to install the Secure Shell server software.
- Chapter 3* Describes how to configure and manage the Secure Shell server and client software.
- Chapter 4* Describes how to use Secure Shell commands.

Reader's Comments

HP welcomes any comments and suggestions you have on this and other Tru64 UNIX manuals.

You can send your comments in the following ways:

- Fax: 603-884-0120 Attn: UBPG Publications, ZKO3-3/Y32
- Internet electronic mail: readers_comment@zk3.dec.com

A Reader's Comment form is located on your system in the following location:

```
/usr/doc/readers_comment.txt
```

Please include the following information along with your comments:

- The full title of the manual and the order number. (The order number appears on the title page of printed and PDF versions of a manual.)

- The section numbers and page numbers of the information on which you are commenting.
- The version of Tru64 UNIX that you are using.
- If known, the type of processor that is running the Tru64 UNIX software.

The Tru64 UNIX Publications group cannot respond to system problems or technical support inquiries. Please address technical questions to your local system vendor or to the appropriate HP technical support office. Information provided with the software media explains how to send problem reports to HP.

Conventions

This manual uses the following typographical conventions:

\	A backslash at the end of a line in an example indicates continuation.
#	A number sign represents the system prompt when you are logged in to a Tru64 UNIX system using the root user account.
net stop	Bold courier type indicates user input.
>>>	The console subsystem prompt is three right angle brackets.
<i>file</i>	Italic (slanted) type indicates variable values, placeholders, and function argument names.
[]	In syntax definitions, brackets indicate items that are optional and braces indicate items that are required. Vertical bars separating items inside brackets or braces indicate that you choose one item from among those listed.
{ }	
. . .	In syntax definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.
cat(1)	A cross-reference to a reference page includes the appropriate section number in parentheses. For example, cat(1) indicates that you can find information on the cat command in Section 1 of the reference pages.
[Ctrl/x]	This symbol indicates that you hold down the first named key while pressing the key or mouse button that follows the slash. In examples, this key combination is enclosed in a box (for example, [Ctrl/C]).

Secure Shell Overview

The Secure Shell software is client/server software that provides a suite of secure network commands that you can use in addition to or in place of traditional nonsecure network commands. Table 1–1 describes the traditional nonsecure network commands and the equivalent Secure Shell command.

Table 1–1: Secure Shell Commands

To	Traditional Command	Secure Shell Command
Execute commands on a remote system	rsh	ssh2
Log in to a remote system	rlogin or telnet	ssh2
Transfer files between systems	rcp or ftp	scp2 or sftp2

See Chapter 4 for more information on using the Secure Shell commands.

The Secure Shell commands create a secure connection between systems running the Secure Shell server and client software by providing:

- **Authentication**
Secure Shell servers and clients use an authentication method to reliably determine each other's identity, then the user's identity.
- **Data encryption**
Secure Shell servers and clients exchange encrypted data. Data encryption is transparent to users.
- **Data integrity**
Secure Shell servers and clients detect whether or not data was intercepted and modified while in transit.
- **Nonrepudiation**
Systems can prove the origin of data so that a user or entity cannot deny having performed a particular action related to data or proof of ownership.

This chapter describes:

- Secure Shell servers

- Secure Shell clients
- Secure Shell server and client communication

1.1 The Secure Shell Server

A Secure Shell server is a system on which the Secure Shell server software is installed and the Secure Shell `sshd2` daemon is started. The Compaq Secure Shell software includes the Secure Shell server software that runs on a system running the Tru64 UNIX Version 5.1A or higher operating system software. The Compaq Secure Shell software is based on SSH Version 2.4.1. software.

The remainder of this guide refers to a Secure Shell server as server.

1.2 The Secure Shell Client

A Secure Shell client is a system on which the Secure Shell client software is installed. The Compaq Secure Shell software includes the Secure Shell client software that runs on a system running the Tru64 UNIX Version 5.1A or higher operating system software.

The Secure Shell client software provides:

- The `scp2` and `sftp2` commands to copy files to and from a server.
- The `ssh2` command to log in and execute commands on a server.
- Other Secure Shell commands to manage the Secure Shell client software.

Optionally, on a Secure Shell client you can configure the `rsh`, `rlogin`, and `rcp` commands and applications that use the `rcmd()` function to automatically use a Secure Shell connection. See Section 3.2.1 for more information.

The remainder of this guide refers to a Secure Shell client as client.

1.3 Server and Client Communication

When the server is started, the `sshd2` daemon listens on port 22 (by default) for a client to initiate a socket connection. When a client connects, the `sshd2` daemon starts a child process. The child process initiates a public host key exchange with the client. The public host key exchange is a process in which the client and server exchange their public host keys to authenticate their identity to each other. A public host key is created on the server as `/etc/ssh2/hostkey.pub` when you install the Secure Shell software.

The first time a client connects to a server, the user is (by default) prompted to accept a copy of the server's public host key. If the user accepts the key,

a copy of the server's public host key is copied to the user's `hostkeys` directory on the client. The client uses this public host key to authenticate the server on subsequent connections. You can also copy the server's public host key in advance to the user's `hostkeys` directory on the client as `key_port_servername.pub`. For example, if the server name is orange, copy its key as `key_22_orange.pub`.

After the client and server authenticate each other, the child process attempts to authenticate the user. The user must have a valid user account and home directory on the server. If the child process fails to authenticate the user, it will refuse the connection. Secure Shell uses one of the following user authentication methods:

- Password
- Public key
- Host-based (Clients and servers can use host-based authentication only if both systems are running the UNIX operating system software.)

See Section 3.3 for information on configuring user authentication methods.

After the child process authenticates the user's identity, the actual connection with the client occurs. The connection includes command execution, encrypted data transfer, and termination of the connection. Once Secure Shell is determined to be used for the connection, all authentication and communication between the client and server will use the Secure Shell connection. When the connection is terminated, the child process started by the `sshd2` daemon terminates.

2

Installing the Secure Shell Software

The Tru64 UNIX Secure Shell software provides the software to run the server and client on a system running the Tru64 UNIX software.

This chapter describes:

- How to download the Secure Shell software
- How to upgrade the Secure Shell software
- How to install the Secure Shell software
- How to install and view the Secure Shell documentation
- How to start the server
- Secure Shell directories and files
- How to deinstall the Secure Shell software
- How to remove the Secure Shell documentation

2.1 Downloading the Secure Shell Software

The Secure Shell software is provided in a compressed tar file that contains:

- A subset called `SSHBASE110` that contains the Secure Shell software.
- A subset called `SSHDOC110` that contains Secure Shell reference pages.
- A tar file called `ssh_ssb_docs.tar.gz` that contains the Secure Shell HTML documentation library.

The Secure Shell software is available from the Compaq Web site at <http://tru64unix.compaq.com/internet>.

To download the Secure Shell software:

1. Create a directory called `/usr/share/doclib/ssh` and change to that directory.
2. Using a Web browser, go to <http://tru64unix.compaq.com/internet> and choose download software.
3. Choose to download the Secure Shell software kit and follow the instructions on the screen.
4. Uncompress the tar file:

```
# gunzip tar_file.tar.gz
```

5. Extract the contents of the tar file:

```
# tar -xvf tar_file.tar
```

2.2 Upgrading the Secure Shell Software

To upgrade the Secure Shell Version 1.0 software, you must deinstall previously installed Secure Shell subsets and install the new Secure Shell software. For example, to deinstall the Secure Shell Version 1.0 software, enter:

```
# /usr/sbin/setld -d SSHBASE100 SSHDOC100
```

2.3 Installing the Secure Shell Software

Installing the Secure Shell software:

- Creates the `/etc/ssh2` directory.
- Creates a public and private host key for the server.
- Modifies the `/etc/services` file and the `/etc/clua_services` file in a TruCluster Server environment to reserve port 22 for the `sshd2` daemon.
- Symbolically links the `/etc/ssh2/knownhosts` directory to the `/var/ssh2/knownhosts` directory.

Use the Tru64 UNIX `setld` command to install the subsets that contain the Secure Shell software and reference pages on a system running the Tru64 UNIX Version 5.1A or higher operating system software. To install the subsets:

1. Change to the directory where the Secure Shell software was downloaded.
2. As the root user, install the subsets:

```
# setld -l . SSHBASE110 SSHDOC110
```

2.4 Installing and Viewing the Secure Shell Documentation

To install the Secure Shell documentation:

1. If necessary, change to the directory where you downloaded the Secure Shell software.
2. Uncompress the documentation tar file:

```
# gunzip ssh_ssb_docs.tar.gz
```
3. Extract the contents from the documentation tar file:

```
# tar -xvf ssh_ssb_docs.tar
```

To view the documentation, open the `/usr/share/doc-clib/ssh/DOCS/HTML/LIBRARY.HTM` file in a web browser.

2.5 Starting the Server Software

After you install the Secure Shell software, you must start the `sshd2` daemon to start the server. When the `sshd2` daemon starts, it uses the values assigned to keywords in the `/etc/ssh2/sshd2_config` file to configure how the server responds to clients.

Review the keywords in the `sshd2_config` file before starting the `sshd2` daemon. See Section 3.1 and `sshd2_config(4)` for more information about the `sshd2_config` file.

To start the `sshd2` daemon, enter:

```
# /sbin/init.d/sshd start
```

In a TruCluster Server environment, you must start the `sshd2` daemon on each cluster member on which you want the `sshd2` daemon to run.

The `sshd2` daemon will automatically start when the system boots.

2.6 Secure Shell Directories and Files

The following files and subdirectories are created in the `/etc/ssh2` directory on the server when you install the Secure Shell software:

- The client configuration file (`ssh2_config`).
- The server configuration file (`sshd2_config`).
- The server private host key file (`hostkey`) and public host key file (`hostkey.pub`).
- A random seed file (`random_seed`). This file contains random data that is used to generate pseudorandom numbers for cryptographic operations.
- Public host key files for the clients with which the server communicates when using host-based authentication are in the `knownhosts` subdirectory.

The `sshd2` daemon is in the `/usr/sbin` directory.

The Secure Shell commands are in the `/usr/bin` directory.

2.7 Deinstalling the Secure Shell Software

To deinstall the Secure Shell software, enter:

```
# setld -d SSHBASE110 SSHDOC110
```

The Secure Shell directories and software subsets are left in place.

2.8 Removing the Secure Shell Documentation

To remove the Secure Shell documentation, enter:

```
# rm -rf /usr/share/doc/lib/ssh/DOCS
```

3

Configuring and Managing the Secure Shell Software

You configure how servers and clients communicate by setting values to keywords in the `/etc/ssh2/sshd2_config` file for the server and in the `/etc/ssh2/ssh2_config` file for the client.

The `/etc/ssh2/sshd2_config` file and the `/etc/ssh2/ssh2_config` file contain keyword-argument pairs, one per line. Keywords are assigned default values, which you can change. Keywords are case insensitive. Empty lines and lines starting with a number sign (`#`) are ignored as comments.

This chapter describes:

- How to configure the server
- How to configure the client
- How to configure user authentication
- Managing the server

3.1 Configuring the Server

The `/etc/ssh2/sshd2_config` file (or the file specified with `sshd2 -f` command) contains configuration keywords and values that the `sshd2` daemon reads when it starts. You can override the values for a keyword in the `sshd2_config` file by entering the keyword and value on the command line when you start the `sshd2` daemon; however, values set this way are reset to the value in the `/etc/ssh2/sshd2_config` file when the `sshd2` daemon restarts.

If you modify the `sshd2_config` file while the `sshd2` daemon is running, you must reset the `sshd2` daemon to implement the change. Changes made this way apply only to new client connections. See Section 3.4.1 for information on resetting the `sshd2` daemon.

Example 3-1 is a sample `/etc/ssh2/sshd2_config` file. See `sshd2_config(4)` for a description of each keyword in the `sshd2_config` file.

Example 3–1: Sample sshd2_config File

```
## sshd2_config
## SSH 2.4 Server Configuration File
##

## General

  VerboseMode  no
# QuietMode   yes
  AllowCshrcSourcingWithSubsystems no
  ForcePTYAllocation no
  SyslogFacility AUTH
# SyslogFacility LOCAL7

## Network

  Port        22
  ListenAddress 0.0.0.0
  RequireReverseMapping no
  MaxBroadcastsPerSecond 0
# MaxBroadcastsPerSecond 1
# NoDelay     yes
# KeepAlive   yes
# MaxConnections 50
# MaxConnections 0
# 0 == number of connections not limited

## Crypto

  Ciphers      AnyCipher
# Ciphers      AnyStd
# Ciphers      AnyStdCipher
# Ciphers      3des
  MACs         AnyMAC
# MACs         AnyStd
# MACs         AnyStdMAC
# RekeyIntervalSeconds 3600

## User

  PrintMotd    yes
  CheckMail    yes
  UserConfigDirectory "%D/.ssh2"
# UserConfigDirectory "/etc/ssh2/auth/%U"
  UserKnownHosts yes
# LoginGraceTime 600
# PermitEmptyPasswords no
```

Example 3–1: Sample sshd2_config File (cont.)

```
# StrictModes    yes

## User public key authentication

HostKeyFile     hostkey
PublicHostKeyFile hostkey.pub
RandomSeedFile  random_seed
IdentityFile    identification
AuthorizationFile authorization
AllowAgentForwarding yes

## Tunneling

AllowX11Forwarding yes
AllowTcpForwarding yes
# AllowTcpForwardingForUsers sjl, cowboyneal@slashdot.org
# DenyTcpForwardingForUsers "2[:isdigit:]*4, peelo"
# AllowTcpForwardingForGroups privileged_tcp_forwarders
# DenyTcpForwardingForGroups coming_from_outside

## Authentication
## Hostbased and PAM are not enabled by default.

# BannerMessageFile      /etc/ssh2/ssh_banner_message
# BannerMessageFile      /etc/issue.net
PasswordGuesses 1
# AllowedAuthentications hostbased,publickey,password
# AllowedAuthentications publickey,pam-1@ssh.com
# AllowedAuthentications hostbased,publickey,password
# RequiredAuthentications publickey,password
# SshPAMClientPath ssh-pam-client

## Host restrictions

# AllowHosts    localhost, foobar.com, friendly.org
# DenyHosts    evil.org, aol.com
# AllowSHosts    trusted.host.org
# DenySHosts    not.quite.trusted.org
# IgnoreRhosts    no
# IgnoreRhosts    no
# IgnoreRootRHosts no
# (the above, if not set, is defaulted to the value of IgnoreRHosts)

## User restrictions
```

Example 3–1: Sample sshd2_config File (cont.)

```
# AllowUsers    "sj*,s[:isdigit:]##,s(jl|amza)"
# DenyUsers    skuuppa,warezdude,31373
# DenyUsers    don@untrusted.org
# AllowGroups  staff,users
# DenyGroups   guest
# PermitRootLogin  nopwd
# PermitRootLogin  yes

## SSH1 compatibility

# Ssh1Compatibility
# Sshd1Path

## Chrooted environment

# ChRootUsers  ftp,guest
# ChRootGroups  guest

## subsystem definitions

subsystem-sftp                sftp-server
```

3.2 Configuring the Client

The `/etc/ssh2/ssh2_config` file contains configuration keywords and values that the client software reads when it starts. Each user can also have their own `$HOME/.ssh2/ssh2_config` file, where `$HOME` is the name of the user's home directory. The `/etc/ssh2/ssh2_config` file is read first, then the user's version is read. The last obtained value for a keyword is used, except for the `EnforceSecureRutils` keyword. See Section 3.2.1 for more information about the `EnforceSecureRutils` keyword.

Example 3–2 is a sample `/etc/ssh2/ssh2_config` file. See `ssh2_config(4)` for a description of each keyword in the `ssh2_config` file.

Example 3–2: Sample ssh2_config File

```
## ssh2_config
## SSH 2.0 Client Configuration File
##
```

Example 3–2: Sample ssh2_config File (cont.)

```
## The "*" is used for all hosts, but you can use other hosts as
## well.
*:

## COMPAQ Tru64 UNIX specific
# Secure the r* utilities (no, yes)
    EnforceSecureRutils          no

## General

    VerboseMode    no
# QuietMode      yes
# DontReadStdin  no
# BatchMode      yes
# Compression    yes
# ForcePTYAllocation yes
# GoBackground   yes
# EscapeChar     ~
# PasswordPrompt "%U%H's password: "
    PasswordPrompt "%U's password: "
    AuthenticationSuccessMsg yes

## Network

    Port          22
    NoDelay       no
    KeepAlive     yes
# SocksServer    socks://mylogin@socks.ssh.com:1080/203.123.0.0/16,198.74.23.0/24

## Crypto

    Ciphers        AnyStdCipher
    MACs           AnyMAC
    StrictHostKeyChecking          ask
# RekeyIntervalSeconds 3600

## User public key authentication

    IdentityFile  identification
    AuthorizationFile authorization
    RandomSeedFile random_seed

## Tunneling
```

Example 3–2: Sample ssh2_config File (cont.)

```
# GatewayPorts    yes
# ForwardX11      yes
# ForwardAgent    yes

# Tunnels that are set up upon logging in

# LocalForward    "110:pop3.ssh.com:110"
# RemoteForward   "3000:foobar:22"

## SSH1 Compatibility

    Ssh1Compatibility          yes
    Ssh1AgentCompatibility     none
# Ssh1AgentCompatibility     traditional
# Ssh1AgentCompatibility     ssh2
# Ssh1Path            /usr/local/bin/ssh1

## Authentication
## Hostbased is not enabled by default.

# AllowedAuthentications     hostbased,publickey,password
    AllowedAuthentications     publickey,password

# For ssh-signer2 (only effective if set in the global configuration
# file, usually /etc/ssh2/ssh2_config)

# DefaultDomain    foobar.com
# SshSignerPath    ssh-signer2

## Examples of per host configurations

#alpha*:
# Host            alpha.oof.fi
# User            user
# PasswordPrompt  "%U:s password at %H: "
# Ciphers         idea

#foobar:
# Host            foo.bar
# User            foo_user
```

Example 3–2: Sample ssh2_config File (cont.)

3.2.1 Configuring Network Commands to Use Secure Shell

You can configure whether or not the `rsh`, `rlogin`, and `rcp` commands and applications that use the `rcmd()` function automatically use a Secure Shell connection by enabling the `EnforceSecureRutils` keyword in the `/etc/ssh2/ssh2_config` file or in a user's `$HOME/.ssh2/ssh2_config` file. By default, the `EnforceSecureRutils` keyword is disabled. If the `EnforceSecureRutils` keyword is enabled in the `/etc/ssh2/ssh2_config` file, it is enabled for all users regardless if it is disabled in user's `$HOME/.ssh2/ssh2_config` file. If the `EnforceSecureRutils` keyword is disabled in the `/etc/ssh2/ssh2_config` file, a user can enable it in their `$HOME/.ssh2/ssh2_config` file.

The following are considerations if you enable the `EnforceSecureRutils` keyword:

- The `sshd` daemon runs and spawns the `srcmd` child process; the `rshd` and `rlogind` daemons do not run.
- The `EnforceSecureRutils` keyword requires that the `AllowTcpForwarding` keyword to be enabled in the `/etc/ssh2/sshd2_config` file, which it is by default. If you enable the `EnforceSecureRutils` keyword, do not disable the `AllowTcpForwarding` keyword.
- The `rsh` and `rcp` commands and applications that use the `rcmd()` function can only use host-based authentication to authenticate users. See Section 3.3.3 for information on host-based authentication.
- The `rlogin` command can use host-based or password user authentication to authenticate users. See Section 3.3.3 for information on host-based authentication. See Section 3.3.1 for information on password authentication.

3.3 Configuring User Authentication

You must configure the client and server to use the same type of user authentication, which can be any or all of the following:

- Password
- Public key

- Host-based (Clients and servers can use host-based authentication only if both systems are running the UNIX operating system software.)

You configure the type of user authentication that the client and server use by assigning values to the `AllowedAuthentications` keyword in the server's `/etc/ssh2/sshd2_config` file and in the client's `/etc/ssh2/ssh2_config` file or user's `$HOME/.ssh2/ssh2_config` file. The user authentication methods are used in the order in which they are listed for the `AllowedAuthentications` keyword. For example, if `hostbased` is listed first, the server will try `hostbased` authentication before trying the next listed authentication. The first successful authentication is the one used. By default, the server tries public key authentication, then password authentication.

3.3.1 Configuring Password Authentication

Password authentication requires that a user have a password-protected user account that can be authenticated by a Tru64 UNIX password authentication method. Tru64 UNIX password authentication methods include:

- BSD
- Network Information Service (NIS)
- Lightweight Directory Access Protocol (LDAP)
- Kerberos
- Enhanced security

To use password authentication, set the value of the `AllowedAuthentications` keyword to include `password`, which is done by default, in the `/etc/ssh2/sshd2_config` file and in the `/etc/ssh2/ssh2_config` file; for example:

```
AllowedAuthentications    password
```

By default, users are allowed only one password attempt. The number of failed password attempts is defined by the value assigned to the `PasswordGuesses` keyword in the `/etc/ssh2/sshd2_config` file.

If you change the value of a keyword in the `/etc/ssh2/sshd2_config` file, you must enter the following command to reset the `sshd2` daemon:

```
# /sbin/init.d/sshd reset
```

When using password authentication, the password prompt will be displayed on `stderr`. If `stderr` is redirected, it may appear that the login command has stopped responding while waiting for the password to be entered.

The way in which users are prompted for their password on a Tru64 UNIX Secure Shell client depends on whether or not a Secure Shell connection is

used. If the server is running the `sshd` daemon, users are prompted as follows:

```
username's password:
```

If the server is not running the `sshd` daemon, users are prompted as follows:

```
Password:
```

3.3.2 Configuring Public Key Authentication

Public key authentication requires that a user have a pair of keys, one called the public key and the other called the private key. These keys are used to authenticate the user and to encrypt and decrypt the data exchanged between clients and servers. The public key is published and distributed to the servers with which the user communicates. The private key is kept secret on the local client and never published or distributed. What one public or private key does, only the other associated public or private key can undo.

Public key authentication requires configuration on the client and on the server.

Note

A user's private key and public key are not the same as the server's private host key (`/etc/ssh2/hostkey`) and public host key (`/etc/ssh2/hostkey.pub`). The server's private and public host keys were created when you installed the Secure Shell software and they are used to authenticate the server. A user creates their private and public keys and they are used to authenticate the user.

3.3.2.1 Configuring Public Key Authentication on the Client

Follow these steps to configure public key authentication on a Tru64 UNIX Secure Shell client:

1. Set the value of the `AllowedAuthentications` keyword in the `/etc/ssh2/ssh2_config` file to include `publickey`, which it does by default; for example:

```
AllowedAuthentications    publickey,password
```

2. Instruct the user to log into their user account and create a public and private key pair by entering the `ssh-pubkeymgr` command. The user will be prompted for the following information:

```

$ ssh-keygen
Setting host to hostname
Checking for publickey authentication to be enabled in the client config..
Your client configuration is all set.

Checking for publickey authentication to be enabled in the server config..
Your server configuration is all set.

Checking for existing user public keys..
Couldn't find your DSA keypair.. I'll generate you a new set..
Running ssh-keygen2... don't forget to give it a passphrase!

Generating 1024-bit dsa key pair
 2 O0o.o0o.o0o.
Key generated.
1024-bit dsa, username@hostname.fqdn, Tue Apr 02 2002 13:40:48 -0500
Passphrase : secret passphrase 1
Again      : secret passphrase
Private key saved to $HOME/.ssh2/id_dsa_1024_a
Public key saved to $HOME/.ssh2/id_dsa_1024_a.pub
If you are logging in from this computer, you need to have an
identification file that defines what private keys will be recognized
when you login. By default, this should be id_dsa_1024_a.

Creating your identity file..

Creating your authorization file..

Creating your local host public key..

The next section allows you to add hosts that you wish to login from using
public key authentication.

Do you want to add any hosts to your authorization file? (Default: yes) yes 2

Type in user and hostname, press return after each one.

Add which user? username
Add which host? hostname
You added username at servername as a trusted login.
Press return to continue or Ctrl-D to exit.
^D
All the new files are in your $HOME/.ssh2 directory.

Now that you have your public keypair generated, you can copy your public
key up to remote hosts so you can login to them using public key
authentication. You also need to add this key, username-hostname.pub,
to the ~/.ssh2/authorization file on the server.

Do you want to upload username-hostname key to a remote host? (Default: yes) 3

```

```

Upload to which host? hostname.fqdn
Which user account? username
Now running scp2 to connect to hostname.fqdn..
Most likely you'll have to type a password :)
Host key not found from database.
Key fingerprint:
xolog-bivic-nomeb-behas-zanet-matuc-hedol-moliv-videl-melal-ciox
You can get a public key's fingerprint by running
% ssh-keygen -F publickey.pub on the keyfile.
Are you sure you want to continue connecting (yes/no)? yes
Host key saved to $HOME/.ssh/hostkeys/key_22_hostname.fqdn.pub
host key for hostname.fqdn, accepted by username Wed Apr 03 2002 11:09:24 -0
500
username@hostname.fqdn's password: password 4
username-hostname.pub | 755B | 0.7 kB/s | TOC: 00:00:01 | 100% 5

Press return to upload to more hosts or Ctrl-D to exit.

```

- 1** Enter a passphrase assigned for the keypair.
- 2** Enter *yes* to add a host entry in the authorization file if you will access your user account on the client from a remote host that uses public host key authentication. A host entry identifies the user name and name of a remote host from which you will access your user account on the client.

If you enter *yes*, you are prompted for the user name by which you want to be authenticated, then for the name of the remote host. Host names should be entered with their fully qualified domain name (fqdn). For example, if the remote host's name is *orange* and its fully qualified domain name is *color.art.com*, enter *orange.color.art.com*

A host entry is added to the authorization file in the following format:

Key username-hostname.pub
- 3** Enter *yes* to copy your public key to a user account on a remote host from which you will access your account on the client. If you enter *yes*, you are prompted for the name of the remote host and the user account to which your public key will be copied. This is usually the user name and name of the remote host for which you added a host entry in the authorization file. Host names should be entered with their fully qualified domain name (fqdn). For example, if the remote host's name is *orange* and its fully qualified domain name is *color.art.com*, enter *orange.color.art.com*
- 4** The user's public key will be copied to the specified user account on the remote host. Users must enter their password for the specified

user account on the remote host because, by default, password authentication is the only authentication available at that time.

- 5 A status message is displayed that shows the results of copying the public key to the remote host.

The `ssh-keygen` command creates:

- A directory called `$HOME/.ssh2` for the user on the client (`$HOME` is the name of the user's home directory)
- The key pair as follows:
 - The `$HOME/.ssh2/id_dsa_1024_a` file contains the user's private key. Only the user for which the key was created should have access this file.
 - The `$HOME/.ssh2/id_dsa_1024_a.pub` file and `$HOME/.ssh2/username-hostname.pub` contains the user's public key. The `username-hostname.pub` is the file that will be copied to servers that use public key authentication and to which the user will connect.
- A file called `$HOME/.ssh2/identification` that contains the following entry that identifies the name of the user's private key file:

```
IdKey id_dsa_1024_a
```
- A file called `$HOME/.ssh2/authorization` that contains the names of public keys for remote hosts from which the user access their user account on the local host.

3.3.2.2 Configuring Public Key Authentication on the Server

To configure public key authentication on the server:

1. Set the value of the `AllowedAuthentications` keyword in the `/etc/ssh2/sshd2_config` file to include `publickey`, which it does by default; for example:

```
AllowedAuthentications publickey,password
```

If you change the value of the `AllowedAuthentications` keyword, you must enter the following command to reset the `sshd2` daemon:

```
# /sbin/init.d/sshd reset
```

2. If the client and server are not the same system, create a `$HOME/.ssh2` directory on the server (`$HOME` is the name of the user's home directory for which public key authentication is being configured.)
3. If necessary, create a file called `authorization` in the user's `$HOME/.ssh2` directory. Add to the `authorization` file a host entry

for the client. A host entry identifies the name of a public key for a client from which the user access their user account on the local host.

A host entry is added to the `authorization` file in the following format:

```
Key username-clienthostname.pub
```

4. If the user did not upload their public key from the client to the remote server when they configured public key authentication, copy from the client the user's public key file (`$HOME/username-hostname.pub`) to the user's `$HOME/.ssh2` directory on the server.

3.3.2.3 Accessing a Remote Server

Example 3–3 shows sample output when logging into a remote server that is using public key authentication.

Example 3–3: Public Key Authentication Login Output

```
$ssh server_name
Passphrase for key "/home/user/.ssh2/id_dsa_1024_a
with comment "1024-bit dsa, created by user@Local
Wed July 19 2001 00:13:43 +0200":
```

3.3.2.4 Restricting User Access

You can restrict a user to only execute certain UNIX commands when they log in to a server that is using public key authentication. To restrict users, enter the UNIX command under the user's `Key` entry in their `authorization` file on the server. For example, the following entry would use public key authentication to authenticate the user, execute the `ls` command in the `log in` directory, then return the user to their local host prompt:

```
Key username-hostname.pub
Command ls
```

3.3.2.5 Managing Passphrases

The passphrase is not the user's Tru64 UNIX user account password. The passphrase is the secret text that the user entered when they entered the `ssh-pubkeymgr` command to create a public and private key pair. The passphrase is used only by the client and server to exchange information about the user.

Users are prompted for their passphrase when they enter a Secure Shell command on a server that uses public key authentication. Users can configure the server so that it does not repeatedly prompt for a user's passphrase during a session by running the Secure Shell agent and loading

their private keys into the agent. When the agent is running, all key-related operations are directed to the agent. The agent terminates when the user logs out or stops the agent.

Follow these steps to run the agent:

1. Log in to the server.
2. Start the agent:

```
ssh-agent2 $SHELL
```

The `$SHELL` environment variable identifies the user's login shell.

Alternatively, users can automatically start the agent when logging in to the server by adding the `ssh-agent2 $SHELL` command to their login file; for example, their `.login` file.

The agent invokes the specified shell as a child process, and the shell prompt appears.

3. Load the private keys into the agent:

```
$ ssh-add2
```

The `ssh-add2` command prompts the user for their passphrase.

3.3.3 Configuring Host-Based Authentication

Host-based authentication is an authentication method that is based on system identification, not password or passphrase identification. Clients and servers can only use host-based authentication if both systems are running the UNIX operating system software.

Host-based authentication requires:

- Users to have a file called `.rhosts` or `.shosts` in their home directory that includes a host name and fully qualified domain name entry for each remote host from which they will access the local host. The `.shosts` is read only by the Secure Shell server. If both files exist, the Secure Shell server reads the `.rhosts` first, then the `.shosts` file. If either of these files allows access for a particular connection, a Secure Shell connection is used, even if the other file forbids it.

Note

Secure Shell requires hosts to use a fully qualified domain name.

- The public host key files for the local host and the remote hosts with which the local host communicates to be in the `/etc/ssh2/knownhosts` directory on the local host.

Because most communication between hosts is reciprocal and a host can be a client and a server; follow these steps on all hosts that will communicate by using host-based authentication:

1. Set the value of the following keywords in the `/etc/ssh2/sshd2_config` file as follows:

- Make sure that the value of the `AllowedAuthentications` keyword includes `hostbased`, which it does by default. If there are other entries, set `hostbased` as the first entry. For example:

```
AllowedAuthentications  hostbased,password
```

- Make sure that the value of the `IgnoreRhosts` keyword is set to `no`. For example:

```
IgnoreRhosts  no
```

If you change the value of a keyword in the `/etc/ssh2/sshd2_config` file, enter the following command to reset the `sshd2` daemon:

```
# /sbin/init.d/sshd reset
```

2. Set the value of the following keywords in the `/etc/ssh2/ssh2_config` file or in a user's `$HOME/.ssh2/ssh2_config` file as follows:

- Make sure that the value of the `AllowedAuthentications` keyword includes `hostbased`. If there are other entries, set `hostbased` as the first entry. For example:

```
AllowedAuthentications  hostbased,password
```

- Set the value of the `DefaultDomain` keyword to be the fully qualified domain name for the local host. For example, if the fully qualified domain name for the local host is `color.art.com`, enter:

```
DefaultDomain  color.art.com
```

3. The local host's public host key must be available in the `/etc/ssh2/knownhosts` directory for other hosts to copy when using host-based authentication. Enter the following command to create a symbolic link to make the local host's public host key available:

```
#ln -sf /etc/ssh2/hostkey.pub \  
/etc/ssh2/knownhosts/hostname.fqdn.ssh-dss.pub
```

The `hostname` is the name of the local host and the `fqdn` is the fully qualified domain name for the local host. For example, if the local host name is `orange` and its fully qualified domain name is `color.art.com`, link as `orange.color.art.com.ssh-dss.pub`

In a TruCluster™ Server environment, the cluster alias public host key must be available in the `/etc/ssh2/knownhosts` directory. Enter the following command to create a symbolic link to make the cluster alias public host key available:

```
# ln -sf /etc/ssh2/hostkey.pub \  
/etc/ssh2/knownhosts/cluster_alias.fqdn.ssh-dss.pub
```

The *cluster_alias* is the name of the cluster alias and the *fqdn* is the fully qualified domain name for the cluster alias.

For a host outside of a TruCluster Server environment to connect to a host inside a TruCluster Server environment, copy the cluster alias public host key file (*/etc/ssh2/hostkey.pub*) to the */etc/ssh2/knownhosts* directory on the host outside of the cluster.

4. Copy the public host key file (*/etc/ssh2/knownhosts/host-name.fqdn.ssh-dss.pub*) from each remote host with which the local host communicates to the */etc/ssh2/knownhosts* directory on the local host.

The *hostname* is the name of the remote host and the *fqdn* is the fully qualified domain name for the host. For example, if the host name is *green* and its fully qualified domain name is *color.art.com*, copy as *green.color.art.com.ssh-dss.pub*

5. In each user's home directory that will use host-based authentication, create a file called *.rhosts* or *.shosts* and add to it an entry for the local host and an entry for each remote host with which the local host communicates. An entry includes the hostname followed by its fully qualified domain name. For example, if the local hostname is *orange* and its fully qualified domain name is *color.art.com*, enter:

```
orange.color.art.com
```
6. The user must be the owner of their *.rhosts* or *.shosts* file in their home directory. Use the *chown* command to set the user as the owner of the *.rhosts* or *.shosts* file and to set the permissions to 600 (read and write by owner only).

3.4 Managing the Server

This section describes how to:

- Start, stop, restart, and reset the *sshd2* daemon
- Restrict users to their home directories
- Create a public and private key
- Forward TCP/IP and X11 data through an Secure Shell connection

3.4.1 Starting, Stopping, Restarting, and Resetting the *sshd2* Daemon

By default, when the *sshd2* daemon starts, it uses the configuration information in the */etc/ssh2/sshd2_config* file. When you start the

sshd2 daemon, you can specify configuration options on the command line. Command line configuration options override values in the `/etc/ssh2/sshd2_config` file and are effective only until the sshd2 daemon restarts. To permanently make a configuration change, edit the `/etc/ssh2/sshd2_config` file. See Section 3.1 for more information on the `/etc/ssh2/sshd2_config` file.

- To start the sshd2 daemon using the configuration information in the `/etc/ssh2/sshd2_config` file, enter:

```
# /sbin/init.d/sshd start
```

To start the sshd2 daemon and change a value for a keyword in the `/etc/ssh2/sshd2_config` file, enter:

```
# /usr/sbin/sshd2 keyword value
```

- To stop the sshd2 daemon, enter:

```
# /sbin/init.d/sshd stop
```
- To restart the sshd2 daemon using the configuration information in the `/etc/ssh2/sshd2_config` file, enter:

```
# /sbin/init.d/sshd restart
```
- To reset the sshd2 daemon, enter:

```
# /sbin/init.d/sshd reset
```

3.4.2 Restricting Users to Home Directories

You use the `ssh-chrootmgr` command to restrict users to their home directory when they use `ssh2` command or the `sftp2` command.

Follow these steps to restrict a user:

1. As the root user, enter the `ssh-chrootmgr` command with the name of the users or groups who you want to restrict. Follow these steps to restrict users:

```
# ssh-chrootmgr username username username
```
2. Edit the `/etc/ssh2/sshd2_config` file and update the value of the `ChRootUsers` entry to include the users (or the `ChRootGroups` entry to include the groups) specified in Step 1.
3. Reset the sshd2 daemon:

```
# /sbin/init.d/sshd2 reset
```
4. Edit the `/etc/passwd` file and configure `/bin/ssh-dummy-shell` as the shell for the users and the users in the groups specified in Step 1.

See `ssh-chrootmgr(1)` and `ssh-dummy-shell(1)` for more information.

3.4.3 Creating a Public and Private Host Key

A public host key and private host key are created with you install the Secure Shell software. You only need to create a new public host key and private host key if you want to change them.

Follow these steps to create a new public host key and private host key:

1. As the root user, stop the `sshd2` daemon:

```
# /sbin/init.d/sshd stop
```
2. Create the host keys:

```
# ssh-keygen2 -P /etc/ssh2/hostkey
```
3. Start the `sshd2` daemon:

```
# /sbin/init.d/sshd start
```

Note

Users who have a copy of the server's old public host key will get a warning message when connecting to a server with a new public host key. Users can delete the server's old public host key file from their `hostkeys` directory on the client. The first time a client connects to the server, the user will be prompted to accept a copy of the server's new public host key.

3.4.4 Forwarding TCP/IP and X11 Data Through a Secure Shell Connection

You can configure TCP/IP ports and X11 connections to forward data by using a Secure Shell connection.

3.4.4.1 TCP/IP Port Forwarding

Forwarding, or tunneling, is a way to forward otherwise insecure TCP data through a Secure Shell connection. For example, you can configure POP3, SMTP, and HTTP connections to use a Secure Shell connection.

Follow these steps to configure TCP/IP port forwarding:

1. Set the value of the `AllowTcpForwarding` keyword to `yes` in the `/etc/ssh2/sshd2_config` file, which is the default.
If you change the value of the `AllowTcpForwarding` keyword, you must enter the following command to reset the `sshd2` daemon:

```
# /sbin/init.d/sshd reset
```

2. Configure the port to forward. There are two kinds of ports you can forward:
 - Local port forwarding (also known as outgoing tunnels)

Local port forwarding forwards data going to a local port to a specified remote port. For example, entering the following command forwards all data going to the specified *local_port* on the local system to the specified *remote_port* on the remote system:

```
ssh2 -L local_port:remote:remote_port user@remote
```
 - Remote port forwarding (also known as incoming tunnels)

Remote port forwarding forwards data going to a remote port to a specified local port. For example, entering the following command forwards all data going to the specified *remote_port* on the remote system to the specified *local_port* on the local system:

```
ssh2 -R remote_port:local:local_port user@remote
```

3.4.4.2 X11 Forwarding

Follow these steps to enable X11 forwarding:

1. Set the value of the `ForwardX11` keyword to `yes` in the `/etc/ssh2/sshd2_config` file, which is not the default.

If you change the value of the `ForwardX11` keyword, you must enter the following command to reset the `sshd2` daemon:

```
# /sbin/init.d/sshd reset
```
2. Test the connection by logging in to the client and enter an X11 command. For example, to start the X clock program, enter:

```
xclock &
```

If the X clock window is displayed, X11 forwarding is working.

Note

You do not need to set the `DISPLAY` variable on the client. Doing so will disable encryption. (X connections forwarded through Secure Shell use a special local display setting.)

Using the Secure Shell Commands

This chapter describes how to use Secure Shell commands to:

- Copy files between clients and servers
- Log in and execute commands on a server

Note

When a client connects to a server for the first time, the user is prompted to accept a copy of the server's public host key. If the user accepts the key, a copy of the server's public host key is copied to the user's `hostkeys` directory on the client. The client uses this public host key to authenticate the server on subsequent connects.

4.1 Copying Files

You can use the following Secure Shell command to copy files:

- The `scp2` command
- The `sftp2` command

4.1.1 Using the `scp2` Command

You can use the `scp2` or `scp` command on a client to copy files to and from a server. The installation process creates a symbolic link from the `scp` command executable to the `scp2` command executable.

The `scp2` command runs with normal user privileges. The basic syntax for the `scp2` command to copy from a local system to a remote system is:

```
scp2 /directory/file user@system:/directory/file
```

The basic syntax for the `scp2` command to copy from a remote system to a remote system is:

```
scp2 user@system:/directory/file user@system:/directory/file
```

Relative paths can also be used; they are interpreted relative to the user's home directory.

See `scp2(1)` for more information about the `scp` command.

4.1.2 Using the sftp2 Command

You can use the `sftp2` command or the `sftp` command on a client to copy files to and from a server. The installation process creates a symbolic link from the `sftp` command executable to the `sftp2` command executable.

The `sftp2` command functions like the `ftp` command, but does not use the `ftp` daemon or the `ftp` client for its connections. The `sftp2` command runs with normal user privileges.

The basic syntax for the `sftp2` command is:

```
sftp2 [options] hostname
```

You can also use the `scp` syntax with the `sftp2` command. See `sftp2(1)` for more information about the `sftp2` command.

4.2 Logging In and Executing Commands on a Server

You can use the `ssh2` command or the `ssh` command on the client to securely log in and execute commands on a server. The installation process creates a symbolic link from the `ssh` command executable to the `ssh2` command executable.

The basic syntax for the `ssh2` command is:

```
ssh2 [options] server_name [command]
```

When a user successfully logs in to a server, the `sshd2` daemon:

1. Changes to run with user privileges.
2. Sets up a basic environment.
3. Changes to the user's home directory.
4. Runs the user's shell.

See `ssh2(1)` for more information about the `ssh2` command.

Index

D

documentation

- installing, 2-2
- removing, 2-4

downloading

- Secure Shell, 2-1

I

installing

- documentation, 2-2
- subsets, 2-2

S

Secure Shell

- client, 1-2
- commands, 1-1t, 4-1
- communication, 1-2
- configuring, 3-1
- configuring commands to use, 3-7
- configuring host-based user authentication, 3-14
- configuring password user authentication, 3-8
- configuring public key user authentication, 3-9
- configuring user authentication, 3-7
- creating public and private host keys, 3-18

- deinstalling, 2-3
 - directories, 2-3
 - downloading, 2-1
 - files, 2-3
 - forwarding ports, 3-18
 - forwarding TCP/IP port, 3-18
 - forwarding X11, 3-19
 - installing, 2-1
 - managing passphrases, 3-13
 - managing server, 3-16
 - overview, 1-1
 - resetting the daemon, 3-16
 - restarting the daemon, 3-16
 - restricting users, 3-17
 - scp2 command, 4-1
 - server, 1-2
 - sftp2 command, 4-2
 - ssh2 command, 4-2
 - starting the daemon, 3-16
 - stopping the daemon, 3-16
 - upgrading, 2-2
 - user authentication, 3-7
- Secure Shell client**
- configuring, 3-4
- Secure Shell server**
- configuring, 3-1
- starting**
- daemon, 2-3
 - server, 2-3
- subsets**
- installing, 2-2