HP-UX Secure Shell A.04.00.000/001 Release Notes

HP-UX 11.0, 11i v1, and 11i v2

Manufacturing Part Number : T1471-90015
E0505

USA

Legal Notices

The information contained herein is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty

A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

U.S Government License

Proprietary computer software. Valid license from HP 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.

Trademark Notices

UNIX is a registered trademark in the United States and other countries, licensed exclusively through The Open Group.

Contents

Contents

Contents

Contents

HP-UX Secure Shell A.04.00.000/001

Information in this document applies to the Web release of HP-UX Secure Shell A.04.00.000/001 (A.04.00).

Secure Shell Versions on HP-UX

Table 1 lists the Secure Shell releases available for HP-UX 11.0, 11i version 1.0, and 11i version 2.0.

Table 1 Secure Shell on HP-UX

Version	Supported Operating System
HP-UX Secure Shell Version A.04.00.000	HP-UX 11.0 HP-UX 11i version 1.0
HP-UX Secure Shell Version A.04.00.001	HP-UX 11i version 2.0

Product Overview

HP-UX Secure Shell A.04.00 is based on the public domain OpenSSH 4.0p1. The client-server architecture supports the SSH-1 and SSH-2 protocols and provides secured remote login, file transfer, and remote command execution. This version of Secure Shell is available on HP-UX 11.0, 11i version 1.0, and 11i version 2.0 platforms.

HP-UX Secure Shell uses hashing to ensure data integrity. HP-UX Secure Shell also provides secure tunneling features, port forwarding, and an SSH agent to maintain private keys on the client.

Following lists the authentication methods supported by HP-UX Secure Shell:

- GSSAPI (Kerberos 5) authentication
- Public key authentication
- Host-based authentication (with public key)
- Challenge-Response authentication (also referred to as Keyboard-Interactive)
- Password authentication

Following lists the technologies tested with HP-UX Secure Shell:

- Kerberos 5/GSSAPI
- OpenSSL
- IPv6
- Trusted Systems
- TCP Wrappers
- PAM (`PAM_UNIX`, `PAM_Kerberos`, `PAM_LDAP`)

Following lists the libraries that HP-UX Secure Shell includes:

- `zlib v1.2.2`
- `OpenSSL v0.9.7e`
- `TCP Wrappers v7.6`

HP supports HP-UX Secure Shell at no additional cost to customers with HP-UX support agreements.

New Features

Following lists the features introduced in HP-UX Secure Shell A.04.00 since the previous release of HP-UX Secure Shell (A.03.91):

The subsequent sections contain a detailed description of the new features.

Address Binding for Port Forwarding Connections

Port forwarding is the process of sending traffic to a specific port (+binding address) on the destination machine. Port forwarding of a message constitutes the following steps:

1. Sending the message to a local port (+address)
2. Forwarding the message to the intended address on the destination machine via an agent such as ssh client or server.

Port forwarding ensures secure communication over an otherwise insecure channel without the client application (or the server) having to modify code (for example, a telnet client may use Secure Shell for secure communication to a telnetd server).

The client provides command-line options (`-DLR`) for local, remote, and dynamic port forwarding.

Starting with this release of Secure Shell, changes have been made to the local, and remote port forwarding connections, so that `ssh(1)` can now allow users to specify addresses that can bind in to port forwarding connections.

Local Port Forwarding

The client can now select one specific interface address for which, incoming requests will be forwarded. This feature provides the client more control, and more security.

For example, to restrict incoming requests to address 10.2.3.2 port 25, the client forwards only the traffic coming in to address 10.2.3.2 (port 25). Use the -L option as shown in the following command:

```
#ssh -L  10.2.3.2:25:10.2.2.3:25 + the rest of the command
syntax
```

Corresponding to the `-L` extensions, HP-UX Secure Shell A.04.00 has also extended the specification of the `LocalForward ssh_config` parameter port on the local host to include the `bind_address`.

Remote Port Forwarding

The server can now select one specific interface address for which incoming requests will be forwarded. This feature provides the server more control, and more security.

For example, to restrict incoming requests to address 10.2.3.1, the server forwards only the traffic coming in to address 10.2.3.1. To set the server up for that kind of remote port forwarding, the client must connect to the server as shown in the following syntax:

```
#ssh -R 10.2.3.1:25:10.2.2.4:25 + <remotehost>
```

NOTE The `bind_address`(10.2.3.1:25) must be local to the remotehost.

Corresponding to the -R extensions, HP-UX Secure Shell A.04.00 has also extended the specification of the RemoteForward ssh_config parameter to include the bind_address.

IMPORTANT

Unlike remote port forwarding, local port forwarding requires only a minor change to the -L syntax.

This is because of the existing GatewayPorts sshd_config directive. Below is a listing of the different parameters of GatewayPorts and the corresponding actions of the server.

NO (default)	The server restricts remote port forwarding to local requestors only, regardless of a binding address specified with the -R option.
YES	The server allows incoming requests on all its network interfaces, regardless of a binding address specified with the -R option.
clientspecified	The server honors the binding address specified for remote port forwarding.

The clientspecified option is introduced in HP-UX Secure Shell A.04.00. See “Remote Binding Control” on page 13 for more information about the client specified option.

Remote Binding Control

To control remote bindings while retaining backward compatibility, the GatewayPorts configuration directive of the sshd daemon has been extended.

The GatewayPorts sshd_config directive now includes a clientspecified option. The clientspecified option, if specified, the server honors the binding address specified for remote port forwarding. This binding address can be specified in line as part of the SSH command line (with the -R option). Alternatively, the option can be defined as part of the RemoteForward sshd_config directive.

Hashing of Host Names and Addresses

Client binaries ssh(1), ssh-keyscan(1), and ssh-keygen(1) now support the hashing of host names and addresses in the known_hosts file. The known_hosts file is located on all client systems, which have ever established an SSH connection with a server or have used ssh-keyscan to retrieve server host information. The known_hosts file keeps track of names and host keys for all servers visited by the user through ssh(1). Users can also direct the output of ssh-keyscan(1) to the known_hosts file.

Until this version of HP-UX Secure Shell, the known_hosts file was written in clear text, and it was visible to all users on the client system. This is a privacy issue, so HP-UX Secure Shell A.04.00 now has HashKnownHosts, a new ssh_config directive. If HashKnownHosts is set to YES (the default option), the host name information retrieved by the client is hashed, and it is written to the known_hosts file.

The client binaries ssh_keyscan(1) and ssh_keygen(1), now support a new option -H. When the -H option is specified with ssh_keyscan(1), host name information retrieved by ssh_keyscan(1) is also hashed. When the -H option is specified with ssh_keygen(1), it hashes all the currently unhashed host information in the specified known_hosts file.

NOTE When used with the -H option, ssh-keygen(1) leaves the already hashed known_hosts entries alone.

A typical known_hosts entry (before hashing) is as follows:

```
ssh1100,172.16.1.163          ssh-rsa
........xpGyYlCu5DxNPGshkJxWPTLLGsYOX7/MpNSs=......
```

Starting with HP-UX Secure Shell A.04.00, the corresponding hashed entry is as follows:

```
|1|hGdAXkx3U/QdwwA0BJbofBlmBRU=|d7tgQAk+q/zyUhZRN9xZ5VN9L9E=

                ssh-rsa
.........xpGyYlCu5DxNPGshkJxWPTLLGsYOX7/MpNSs=.....
```

- To hash an existing file, use the following command:

  ```
  #ssh-keygen -H [ -f known_hosts file ]  // Assume
  known_hosts file is "/.ssh/known_hosts"
  ```

You see the following output:

```
/.ssh/known_hosts updated.
      Original contents retained as /.ssh/known_hosts.old
      WARNING: /.ssh/known_hosts.old contains unhashed
entries
      Delete this file to ensure privacy of hostnames
...........
```

- To retrieve host keys for a specific host and then hash the host name, use the following command:

```
#ssh-keyscan  -H  hostname
```

Includes New ssh-keygen(1) Options

New `ssh-keygen` options for managing keys in `known_hosts` files, are introduced in HP-UX Secure Shell A.04.00. The `ssh-keygen` utility now has the ability to:

- Search for hosts by name
- Convert an unhashed `known_hosts` file into one with hashed names
- Delete hosts by name

Table 2 lists the new options added to ssh-keygen.

Table 2 New Options in ssh-keygen

Option	Description
-F hostname[-f known_hosts_file]	Searches for the specified hostname in a known_hosts file, and lists any occurrences found. This option can be used to find hashed host names or addresses. The option can also be used in conjunction with the -H option to print found keys in a hashed format. To search for a host name in a (hashed) known_hosts file, use the following command: #ssh-keygen -F localhost The output of this command is as follows: Host localhost found: line 1 type RSA1 |1|AK7Ea+ekY5zep0xzhrjamtsDETE=|5STclD07 U2yxU8sqNM6kaFDnd+8= 1024 35 1406216..... Host localhost found: line 2 type RSA |1|f+Qcomb/S2CJ4bBW9nnJ/iK7nhA=|80bhDySi IfniB3iMzcbVei62ddo= ssh-rsa AAAAB3NzaC1.....
-H [-f known_hosts_file]	Hashes a known_hosts file, and creates two separate files. This option replaces all host names and addresses with hashed representations. These hashes may be used normally by ssh and sshd, but they do not reveal identifying information if the contents of the file are disclosed. This option does not modify existing hashed host names. HP recommends that the option be used on files that mix hashed and unhashed names. To hash the hostnames in a known_hosts file, use the following command: #ssh-keygen -H

Table 2 New Options in `ssh-keygen` (Continued)

Option	Description

`-R hostname`	Removes all keys that belong to a host name from a `known_hosts` file. This option can be used to delete hashed hosts.
	To remove all keys belonging to a host name in a `known_hosts` file, use the following command:
	`#ssh-keygen -R localhost`
	The output of this command is as follows:
	`/.ssh/known_hosts updated.`
	`Original contents retained as /.ssh/known_hosts.old`

Improved Logging of Connection Sources

HP-UX Secure Shell currently supports four directives (`AllowUsers`, `DenyUsers`, `AllowGroups` and `DenyGroups`) to allow or deny access to specific users and groups.

For example, `AllowUsers` is used to allow access only to listed users, while denying access to all other users. When an unlisted user attempts to connect to the server, an error is generated as in the following example:

```
Mar 23 12:46:51 ssh1100 sshd[2672]: User <username> not allowed
because not listed in AllowUsers
```

However, this message contains no information about the client host from which the user made the connection attempt. Starting with HP-UX Secure Shell A.04.00, the error message contains the canonical host name of the client, as in the following example:

```
Mar 23 12:46:51 ssh1100 sshd[2672]: User <username> from
<client's canonical hostname> not allowed because not listed in
AllowUsers
```

Improved Handling of Bad Data in authorized_keys Files

In the previous versions of Secure Shell, Digital Signature Algorithm (DSA) keys (for example,`id_dsa.pub`) with 8192 bits or more were not correctly recognized due to the limitation of buffer size (4096), which holds the key. As a result, key size greater than 4096 bits produced fatal errors. To eliminate this problem, the buffer size has been increased to 8192 bits.

For SSH-2 protocol, lines in the `Pubkey` file are usually several hundred bytes long (because of the size of the public key encoding) up to a limit of 8 kilobytes, which permits DSA keys up to 8 kilobits and Rivest, Shamir, and Adleman (RSA) keys up to 16 kilobits.

Improved Connection Multiplexing Support in ssh(1)

To allow HP-UX Secure Shell control over a running multiplexing master connection, a new “command mode” has been added that enables control of a running multiplexing master connection, including checking that it is up, determining its PID and asking it to exit.

In previous versions of Secure Shell, following a successful authentication, the client would open multiple channels. Each channel handled communication for either a different terminal session or for an X11 session. Now, with this enhancement, the client session can communicate with its master process through a new command-line option (`-O ctl_cmd`).

The `-O ctl_cmd` command-line option is used for controlling an active connection-multiplexing master process. When the `-O` option is specified, the `ctl_cmd` argument is interpreted and passed to the master process. Following lists the valid command values for `ctl_cmd`:

`check`:	check whether the master process is running
`exit`:	send a request to the master process to exit

Output From Failing PAM Session Modules

Secure Shell now sends output from failing PAM session modules to the user before exiting. A Bugzilla reference can be found at the following url:

`http://bugzilla.mindrot.org/attachment.cgi?id=679`

Table 3 describes the differences between HP-UX Secure Shell A.03.91 and HP-UX Secure Shell A.04.00 in handling PAM-related errors.

Table 3 Differences in Handling PAM Related Errors

HP-UX Secure Shell A.03.91	HP-UX Secure Shell A.04.00
Does not revert forwarding mechanisms. For example, if agent forwarding is enabled via the `ssh -A` option, `sshd` fails to disable it before or after the user exits.	Reverts all forwarding mechanisms (port, agent, or X11forwarding) before the user exits.
Does not display login messages. For example, if a user is not allowed to view or access the `/etc/nologin` file, the user will not be able to list the file (using the `ls` command) or view the contents of the file.	Displays both `/etc/nologin` presence and its content to the user.
LogLevel = `FATAL`	LogLevel = `ERROR`

Choice of `AddressFamily` Configuration Directive in `sshd_config` (Server)

The `AddressFamily` `sshd_config` directive, introduced in HP-UX Secure Shell A.04.00, enables the server to restrict users to use a specified address family only. This directive enables global control over IPv4/IPv6 usage. Since the server takes control over the address family to be used, it can process connections irrespective of the address format of the server and the client's actual host.

The `AddressFamily` `sshd_config` directive takes three possible values:

`inet`	support IPv4 address only
`inet6`	support IPv6 address only
`any`	support both IPv4 and IPv6 addresses

`AddressFamily` specifies the address family to be used by `sshd`. Valid arguments are `any`, `inet` (use IPv4 only) or `inet6` (use IPv6 only). The default argument is `any`.

New Configuration Directives in `ssh_config` (Client)

Following are the new configuration directives that `ssh_config` (client) includes:

`HashKnownHosts`

This directive indicates that ssh hashes host names and addresses when they are added to `$HOME/.ssh/known_hosts`. These hashed names are used by `ssh` and `sshd`, but they do not reveal identifying information in case the contents of the file are disclosed. The OpenSSH default is `NO`, but the HP-UX Secure Shell 4.0 default is YES.

The `HashKnownHosts` directive is used to set up the user with more secure options. Users can change this value (or remove this directive) at any time.

`KbdInteractiveDevices`

This directive specifies the list of methods to be used in keyboard-interactive authentication. Multiple method names must be comma separated. The default is to use the server-specified list.

Unsupported Features

Starting with HP-UX Secure Shell A.03.81, the following features are not supported:

- The `KerberosGetAFSToken` option for `sshd(8)`

  This directive specifies whether or not to accept forwarded Andrew File System (AFS) tokens. HP-UX Secure Shell A.04.00 does not support this feature.

- Host keys in DNS (`draft-ietf-secsh-dns-xx.txt`)

  HP-UX Secure Shell A.04.00 does not support this feature.

Defects Fixed in HP-UX Secure Shell A.04.00

All defect fixes included in the previous versions are also available in HP-UX Secure Shell A.04.00. Table 4 lists and describes the defects fixed in this version of Secure Shell.

Table 4 Defect Fixes

JAG ID	Description
JAGaf61054	Fix for a problem where after receiving a `SIGTERM`, `sshd` was not writing the "end of session" record to the `wtmp` file. As a result, the "last" command still shows the session as active. The problem has existed since HP-UX Secure Shell 3.5.
JAGaf61055	Fix for a base code issue where if `UseLogin` is set to `YES`, then each successful login will write two `wtmp` records instead of one.
JAGaf62502	Fix in the `chroot` setup script for HP-UX 11.00. The script was assuming the existence of kerberos-related libraries. The problem been fixed to copy kerberos-related libraries to the `chroot` environment only if they are present on the system.

Known Problems and Workarounds

Following lists the known problems and workarounds in HP-UX Secure Shell A.04.00:

- If a server environment has `UsePAM` set to `YES` and `pam.conf` set to `PAM_LDAP`, then an SSH user connection attempt fails if it is using key-based authentication (for example, Public Key, Host-based key or GSSAPI). The problem occurs in account management. `PAM_LDAP` account management returns an error if the account management call is not preceded by a successful `PAM_LDAP` authentication. For key-based authentication, Secure Shell does the authentication by itself and does not invoke PAM authentication.

  Workaround: There is no good workaround for the problem. If possible, users can choose a password-based authentication mechanism if account management is set up through `PAM_LDAP`.

- In some user environments, public-key based SSH connection fails with `PAM_UNIX`. This is a rare occurrence, and the only known workaround for this problem is to restart the `sshd` daemon.

- The `chroot` functionality does not work if the `UseLogin` directive in `sshd_config` is set to `YES`. By default, `UseLogin` is set to `NO`.

HP-UX Secure Shell and the Strong Random Number Generator

HP-UX Secure Shell requires that a random number generator be located on the system. It searches for `/dev/urandom` and `/dev/random` (in that sequence) on the system and uses the first device it finds. If it fails to locate these two devices, HP-UX Secure Shell uses its own internal random number generator program. The `/dev/urandom` and `/dev/random` devices are available by default on HP-UX 11i v2 systems. These devices can also be obtained for HP-UX 11i v1 by downloading and installing the HP-UX Strong Random Number Generator from `http://software.hp.com`. HP recommends that Secure Shell users on HP-UX 11i v1 systems install the Strong Random Number Generator product as it significantly speeds up program initialization and execution time for some commands.

HP-UX Secure Shell Resources

For more information on Secure Shell, read the following:

- HTML and pdf versions at `http://docs.hp.com` (*Internet and Security Solutions*)
- A README text version in the software at: `/opt/ssh/README.hp`
- The HP Instant Information CD
- OpenSSH at `http://www.openssh.com`
  - — FAQs, Mail List Archives, Security pages, manpages
- IETF at `http://www.ietf.org/` (go to Working Groups > Security)
- The HP book *HP-UX 11i Security* by Chris Wong.
- Secure Shell FAQs at: `http://www.employees.org/~satch/ssh/faq/ssh-faq.html`
- *O'Reilly's SSH, The Secure Shell-The Definitive Guide by Daniel J. Barrett and Richard E. Silverman*.

HP-UX Secure Shell Commands

Table 5 describes the HP-UX Secure Shell commands. Refer to the manpage for each command for more information.

Table 5 HP-UX Secure Shell Commands

Command	Description
`ssh`	Client program similar to `rlogin` and `rsh`
`sshd`	Secure shell server daemon
`sftp`	Secure `ftp` program
`scp`	Secure file copy program similar to `rcp`
`slogin`	Symbolic link to `ssh`
`ssh-agent`	Authentication agent to store private keys
`ssh-add`	Tool for adding keys to `ssh-agent`
`ssh-keygen`	Tool for manually creating public and private keys
`sftp-server`	The `sftp` server subsystem automatically initiated by `sshd`
`ssh-keyscan`	Tool for gathering public host keys

Prerequisites

This section details the prerequisites for installing HP-UX Secure Shell A.04.00.

Patch Requirements

HP has tested HP-UX Secure Shell A.04.00 with the following Support Plus patches. HP recommends that HP-UX 11.0 customers install these Support Plus patches. HP mandates that HP-UX 11i version 1.0 customers must install these Support Plus patches.

Table 6 Quality Packs for HP-UX 11.0 and 11i v1

Operating System	Recommended Support Plus Patch Date / Release # / Part #
HP-UX 11.0	March 2003 SP60 Quality Pack
HP-UX 11i version 1.0	December 2002 Support Plus release / media

The HP-UX 11i v1 (B.11.11) Support Plus release media contains the standard HP-UX patch bundles, which are also available on the HP IT Resource Center Website. If you have the HP-UX 11i v1 (B.11.11) Support Plus release media for December 2002, then you will find the required patches. If you no longer have the media available, then complete the following steps:

1. Go to the IT Resource Center (ITRC): `http://www.itrc.hp.com`
2. Log in to the appropriate site: Americas/Asia-Pacific or European
3. Select maintenance and support (hp products).
4. Select standard patch bundles - find patch bundles
5. Select HP-UX patch bundles

The standard HP-UX patch bundles index page lists the release dates for the current patch bundles. Selecting a specific release date will provide you with a list of all the patch bundles released on that particular date.

NOTE The standard HP-UX patch bundles are cumulative so that if you do not find an older bundle, such as a patch bundle on the Dec'02 Support Plus 11.11 media, you can select the latest 11.11 release and utilize the latest version of that particular patch bundle.

HP recommends that the following `libc` patches be installed for use with HP-UX Secure Shell A.04.00:

Table 7 libc Patches

Operating System Version	Patch
HP-UX 11.0	PHCO_25976
HP-UX 11i version 1.0	PHCO_27740

HP recommends that the following PAM patches be installed for use with HP-UX Secure Shell A.04.00:

Table 8 PAM Patches

Operating System Version	Patch
HP-UX 11.0	PHCO_29249
HP-UX 11i version 1.0	PHCO_30402

HP recommends that the following pthreads patches be installed for use with HP-UX Secure Shell A.04.00:

Table 9 pthreads Patches

Operating System Version	Patch
HP-UX 11.0	PHCO_26960
HP-UX 11i version 1.0	PHCO_26466

System Requirements

Following are the minimum system requirements for installing HP-UX Secure Shell A.04.00:

Operating System

- HP-UX 11.0
- HP-UX 11i version 1.0
- HP-UX 11i version 2.0

Hardware

- HP/9000 Servers and Workstations
- HP Integrity Servers

Disk Space

Approximately 32MB of disk space

Software Availability in Native Languages

This version of HP-UX Secure Shell is available in English only.

NOTE HP has tested HP-UX Secure Shell A.04.00 with HP-UX Secure Shell A.03.91 and expects compatibility with previous versions of HP-UX Secure Shell.

HP-UX Secure Shell Software Availability

HP-UX Secure Shell is available on the following:

- HP Software Depot at `http://www.software.hp.com`
- HP-UX Application Release CDs
- HP-UX 11i version 2.0 Operating Environment (OE)

Installing HP-UX Secure Shell

You do not need to remove previous versions of HP-UX Secure Shell before upgrading to HP-UX Secure Shell A.04.00. However, if you are reverting to an older version of HP-UX Secure Shell, HP recommends that you remove the new product before installing the older version.

To install HP-UX Secure Shell, complete the following steps:

Step 1. Log in as `root`.

Step 2. Insert the software CD into the appropriate drive if installing from the Application Release CD. If installing from `http://software.hp.com`, download the depot and use the `swinstall` directions provided on the Installation page where you downloaded the software.

Step 3. Run `$ swinstall -s <fully-qualified depot source path>`

Step 4. In the `Source Depot Path` field, enter the drive mount point and click **OK**. Change the `Source Host Name` if needed.

Step 5. Select `T1471AA` from the list of available software and click **Mark for Install** on the `Actions` menu.

Step 6. Click **Install** on the `Actions` menu.

Step 7. Click **OK** in the `Install Analysis` window when the `Status` field displays a Ready message.

Step 8. Click **Yes**. The `swinstall` command loads the HP-UX Secure Shell files on the system in approximately 3 to 5 minutes.

NOTE

The `sshd` daemon is preconfigured, and is started after installation.

The `swinstall` command installs HP-UX Secure Shell in the `/opt/ssh/` directory.

Configuring HP-UX Secure Shell

Use the following information as a supplement to the manpages and O'Reilly's *SSH, The Secure Shell -- The Definitive Guide*.

TIP HP recommends that you use SSH-protocol version 2 to eliminate the risk of an insertion attack.

When you install HP-UX Secure Shell A.04.00, RSA1, RSA, and DSA server keys are generated (if these keys did not exist previously). Table 10 describes the host keys that are generated by HP-UX Secure Shell. These keys are located in `/opt/ssh/etc/` after installation.

Table 10 HP-UX Secure Shell Host Keys

Host Key	Description
`ssh_host_key` `ssh_host_key.pub`	RSA1 private and public host keys for SSH-1
`ssh_host_rsa_key` `ssh_host_rsa_key.pub`	RSA private and public host keys for SSH-2
`ssh_host_dsa_key` `ssh_host_dsa_key.pub`	DSA private and public host keys for SSH-2

HP-UX Secure Shell server and client configuration files are available at the locations listed in Table 11 after installation. You can use the default values for the configuration directives listed in these files, or you can modify these values according to your needs. See "Configuration Directive Settings in the sshd_config File" on page 37 for more information.

Table 11 Configuration Files

File	File Location
Server configuration file	`/opt/ssh/etc/sshd_config`

Table 11 Configuration Files (Continued)

File	File Location
Client configuration file	/opt/ssh/etc/ssh_config

Password Authentication

HP-UX Secure Shell uses PAM (Pluggable Authentication Module) for password authentication. The sshd daemon uses its own configuration lines in /etc/pam.conf. You do not need to edit the /etc/pam.conf file prior to using sshd because of the OTHER service in pam.conf. Unspecified applications in /etc/pam.conf always use the OTHER service. You can create a link to sshd and change the service name in /etc/pam.conf to the link name. However, the service name in /etc/pam.conf must match the name of the daemon invoked.

For example, if you create a link to sshd named lsshd and then invoke lsshd, PAM looks for the service name lsshd. For more information on configuring PAM, refer to the *pam.conf(4)* manpage.

HP-UX Secure Shell supports the PAM interface. Starting with HP-UX Version A.03.71, UsePAM was introduced in the sshd_config file. This is the server configuration file that you can use to enable or disable PAM support. The UsePAM directive is enabled by default in the configuration file.

Following is a sample usage of the UsePAM directive:

```
UsePAM yes
UsePAM no
```

The setting in the /etc/pam.conf file determines the PAM module that must be used. When the setting is YES, PAM uses sshd for authentication. When the setting is NO, sshd bypasses PAM and calls getpwnam() directly to authenticate the user.

HP-UX Secure Shell has been tested with HP PAM implementations such as PAM_UNIX, PAM_Kerberos, and PAM_LDAP. HP-UX Secure Shell works with any other PAM module.

Key Generation

By default, HP-UX Secure Shell is set to use SSH-2 only. The default setting for `ssh-keygen` is null. The `ssh-keygen` utility can create SSH-1 (RSA1) and SSH-2 (RSA, DSA) key pairs. Use the `-t` option to generate SSH-2 key pairs. You must specify the type of key that you want to generate. For example, `ssh-keygen -t dsa` generates SSH-2 DSA key pairs. For more information refer to the *ssh-keygen* manpage for more information.

Public Key Authentication

Public key authentication is used to generate key pairs for the user at the client system. The -t option accepts the following three values:

- RSA1- Use this option to generate keys for the SSH-1 protocol. Example:

  ```
  #ssh-client-system ssh-keygen -t rsa1
  ```

- RSA- Use this option to generate keys for the SSH-2 protocol. Example:

  ```
  #ssh-client-system ssh-keygen -t rsa
  ```

- DSA- Use this option to generate keys for the SSH-2 protocol. Example:

  ```
  #ssh-client-system ssh-keygen -t dsa
  ```

NOTE RSA1 and RSA value generate keys in different lengths.

Use the default file name `~/.ssh/identity` for SSH-1, or `~/.ssh/id_rsa` and `~/.ssh/id_dsa` for SSH-2.

Depending on the protocol and algorithm you use, you can securely copy or append the public key files `~/.ssh/identity.pub`, `~/.ssh/id_dsa.pub`, or `~/.ssh/id_rsa.pub` from the client to the server and name it `~/.ssh/authorized_keys`. Append the file using the `cat` command.

Port Forwarding

Port forwarding implies using a local port in place of a remote port. The procedure to invoke `ssh` is based on the need to enable persistent port forwarding.

Following procedure explains how to invoke `sshd` to enable port forwarding.

1. Let `<lport>` be a port on your local system that you have access to. If you are not root your port number must be greater than 1024. Ensure the port is not being used.
2. Let `<rmachine>` reflect the remote system you need to connect to.
3. Let `<rport>` be the remote port on `<rmachine>`.
4. Use the `-L` option to specify port mapping.
5. Use the `sleep` command with a large number to make the connection persistent. For example, if you use the `sleep` command with a 1000000 number, the connection will be persistent for 11 days. If you use `sleep` with a small number like 10, the port forwarding connection lasts for only 10 seconds.

   If the port that is being forwarded is in use, it is not closed, till the connection is terminated. Therefore, if you use a short sleep period and use port forwarding longer than the sleep period, the connection will last as long as it is used and then stop.

   `<command>` runs on the remote machine and cannot have any terminal interaction.
6. Connect to `<lport>` on the localhost to use the port forwarding connection, as shown in the following example:

   ```
   $ ssh -L<lport>:localhost:<rport> <rmachine> <command>
   ```

For example: You want to connect to a service on remote machine B, which listens on port 123. Your application resides locally on machine A and is configured to connect to port 123 on machine B. Start a Secure Shell port forwarding session with the following command:

```
ssh -L22222:localhost:123 machineB sleep 1000000
```

This command configures your application to connect to port 22222 on localhost. The application opens a connection to port 22222 on localhost, where `ssh` waits for a connection. The `ssh` command directs all communication to `sshd` on machine B. This command also opens a

connection to port 123. All communication travels as if there is a direct connection between machine A and machine B. However all communication is through the SSH encrypted connection.

IMPORTANT To use port forwarding on an IPv6-enabled machine running HP-UX Secure Shell, if you do not set the `GatewayPorts` option in `sshd_config` command to `yes`, applications will fail to connect to the specified port.

X11 Forwarding

You can modify `sshd_config` and `ssh_config` to allow X11 forwarding. Edit the `sshd_config` file to set the display instance (default = 10). After a secure connection is established, HP-UX Secure Shell sets the `display` environment variable to the following:

```
<server machine>:<display instance>.0
```

`<server machine>` is the machine you connected to when you initiated the Secure Shell connection. Secure Shell attempts to set `<display instance>` to the display instance specified in the `sshd_config` configuration file. However, if that port is in use, HP-UX Secure Shell sets it to a different number that is greater than the display instance specified. X programs display on the originating machine when started.

IMPORTANT When using X11 forwarding to an IPv6-enabled machine running HP-UX Secure Shell, the `X11UseLocalhost` option sometimes does not work. If the machine has Transport Optional Upgrade Release (TOUR) Version 1.0 or greater installed, `X11UseLocalhost` can be set to either `yes` or `no`. If TOUR is not installed, the `X11UseLocalhost` option must be set to `no` in the `sshd_config` file on the IPv6-enabled machine. X11 applications will not connect to the display if the `X11UseLocalhost` option is not set to `no`.

Set the `X11UseLocalhost` value to `no` if the X11 client uses a pre-X11R6 library for X11 forwarding.

Configuration Directive Settings in the `sshd_config` File

The HP-UX Secure Shell daemon (`sshd`) reads a set of runtime configuration directives from the `sshd_config` file. By default, this file is located in the `/opt/ssh/etc` directory. These configuration directives control `sshd` behavior for various functions. Following sections describe the valid configuration flags, as set by HP-UX Secure Shell.

`AcceptEnv`

Use this directive to specify the environment variables sent by the client to be copied into the session environ. Environment passing is only supported for SSH-2 protocol. You can specify variables by name. Variables can contain the wildcard characters "*" and “?”. Use whitespace to separate multiple environment variables. You can also spread multiple environment variables across multiple `AcceptEnv` directives.

NOTE Some environment variables can bypass restricted user environment. Care should be taken in the use of the `AcceptEnv` directive. The default value is not to accept any environment variables.

`AllowGroups`

Use this directive to allow only users whose primary or supplementary group list matches one of the patterns. This directive can be followed by a list of group name patterns, separated by spaces. The ‘*’ and ‘?’ symbols can be used as wildcards in the patterns. Only group names are valid; a numerical group ID is not recognized. By default, login is allowed for all groups. By default, this directive is not specified in the `sshd_config` file.

`AllowTCPForwarding`

Use this directive to enable or disable TCP forwarding. The default setting is `YES`. Disabling TCP forwarding does not improve security unless users are also denied shell access, as they can always install their own forwarders.

AuthorizedKeysFile

Use this directive to specify the file to be used for public key authentication. The `AuthorizedKeysFile` can contain tokens of the form `%T`, which are substituted during connection setup. The following tokens are defined:

- `%%` is replaced by `%`
- `%h` is replaced by the home directory of the user being authenticated
- `%u` is replaced by the username of the user being authenticated

After expansion, `AuthorizedKeysFile` is taken to be an absolute path or one relative to the home directory of the user. The default is `.ssh/authorized_keys`.

ChallengeResponseAuthentication

Use this directive to enable Challenge-Response (also known as Keyboard-Interactive) authentication. All authentication styles from `login.conf(5)` are supported.

ClientAliveCountMax

Use this directive to specify the number of client alive messages, which can be sent without `sshd` receiving any messages back from the client. If the specified threshold is reached while client alive messages are being sent, `sshd` will disconnect the client and terminate the session. Use of this directive is different from the `KeepAlive` directive. The client alive messages are sent through the encrypted channel and cannot be spoofed. The client alive directive enables a client or server detect if any connection is inactive.

ClientAliveInterval

This directive is protocol version 2 specific and is used to send a request to nonresponsive clients. This directive sets the timeout interval in seconds. If no data is received from the client after the specified timeout interval, `sshd` sends a message through the encrypted channel to request a response from the client.

Compression

Use this directive to compress the data sent over HP-UX Secure Shell connections before it is encrypted. It also decompresses data received on the client side in after it is decrypted at the client side.

DenyGroups

Use this directive followed by a list of group name patterns, separated by spaces to deny login for users whose primary group or supplementary group list matches one of the patterns. The '*' and '?' symbols can be used as wildcards in the patterns. Only group names are valid; a numerical group ID is not recognized. By default, login is allowed for all groups.

DenyUsers

Use this directive followed by a list of user name patterns, separated by spaces to deny login for user names that match one of the patterns. The '*' and '?' symbols can be used as wildcards in the patterns. Only user names are valid; a numerical user ID is not recognized. By default, login is allowed for all users. If the pattern takes the form `USER@HOST`, then `USER` and `HOST` are separately checked, restricting logins to particular users from particular hosts.

GatewayPorts

Use this directive to specify that `sshd` allows remote port forwardings to bind to non-loopback addresses, thus allowing other hosts to connect. Use one of the following arguments with this directive:

- `NO`

  This option forces remote port forwardings to be available to the local host only

- `YES`

  This option forces remote port forwardings to bind to the wildcard address

- `clientspecified`

  This option allows the client to select the address to which the forwarding is bound.

The default option for the GatewayPorts directive is NO.

GSSAPIAuthentication

Use this directive to specify whether GSSAPI can be used to authenticate users.

GSSAPICleanupCredentials

Use this directive to specify if the user credentials must be automatically destroyed on logout.

GSSAPIEnableMitmAttack

Use this directive to specify whether the (deprecated) GSSAPI authentication method is to be enabled for the server. The default setting is NO. If set to YES, older versions of HP-UX Secure Shell clients such as 3.8, and 3.7 can connect to an HP-UX Secure Shell A.04.00 server using GSSAPI authentication. This directive was introduced in HP-UX Secure Shell 3.8, and applies to SSH-2 protocol only.

The new GSSAPI_WITH_MIC authentication method was introduced in HP-UX Secure Shell 3.8, but a patch was provided to maintain compatibility with the previous GSSAPI authentication method. This patch enables older versions of the client to connect to newer versions of the server (using GSSAPI authentication). Similarly, newer versions of the client can now connect to older versions of the server.

The gssapi patch has been preserved in this version of Secure Shell. This configuration directive works exactly the same way as it did in HP-UX Secure Shell 3.8. You must set the GSSAPIEnableMitmAttack directive to YES if you want the server to enable support for the older GSSAPI authentication method.

HostbasedAuthentication

Use this directive to specify whether rhosts or /etc/hosts.equiv authentication combined with successful public key client host authentication is allowed. This directive checks /etc/ssh_known_hosts and $HOME/.ssh/known_hosts for the public key. This directive applies to SSH-Protocol-2. The default setting is NO.

HostKey

Use this directive to specify a file containing a private host key used by HP-UX Secure Shell. By default, this directive specifies the `/opt/ssh/etc/ssh_host_key` file for protocol version 1, and `/opt/ssh/etc/ssh_host_rsa_key` and `/opt/ssh/etc/ssh_host_dsa_key` files for protocol version 2.

NOTE The `sshd` daemon does not use a file if it is group/world-accessible. It is possible to use the `HostKey` directive to specify multiple host key files. The `rsa1` keys are used for SSH protocol version-1 and `dsa` or `rsa` keys are used for SSH protocol version-2.

IgnoreRhosts

Use this directive to specify the usage of `.rhosts` and `.shosts` file in `RhostsRSAAuthentication` and `HostbasedAuthentication`. This directive when enabled ignores the `.rhosts` and `.shosts` files but continues to use the `/etc/hosts.equiv` and `/opt/ssh/etc/shosts.equiv` files.

IgnoreUserKnownHosts

Use this directive to specify whether the `sshd` daemon ignores the `$HOME/.ssh/known_hosts` files when authenticating using `RhostsRSAAuthentication` or `HostbasedAuthentication`. The default setting for this directive is `NO`.

KerberosAuthentication

Use this directive to specify whether the Kerberos KDC will validate the password provided by the user for PasswordAuthentication. To use this option, the server needs a Kerberos `servtab` to verify the KDC identity. This directive is supported only if it is enabled at compile time with the option, `--with-kerberos`. In OpenSSH, the default value of `KerberosAuthentication` is `NO`. In HP-UX Secure Shell, the default value has been modified to `YES`.

KerberosOrLocalPasswd

Use this directive to specify that passwords be validated through other mechanisms such as `/etc/passwd/` when Password Authentication through Kerberos fails. This is useful in an environment where not every user authenticates through Kerberos. The default setting for this directive is `YES`.

KerberosTicketCleanup

Use this directive to specify if the user ticket cache file must be destroyed automatically. When you enable this directive, the user cache file is destroyed automatically when the user logs out. The default setting for this directive is `YES`.

ListenAddress

Use this directive to specify the local addresses that the `sshd` daemon must listen on. You can use one of the following forms with this directive:

- `ListenAddress host|IPv4_addr|IPv6_addr`
- `ListenAddress host|IPv4_addr:port`
- `ListenAddress [host|IPv6_addr]:port`

If a port is not specified, the `sshd` daemon listens on the specified address and all prior specified port options. You can specify multiple `ListenAddress` options. Additionally, any port options must precede this option for non port qualified addresses. The default value of this directive is to listen on all local addresses.

LoginGraceTime

Use this directive to specify the period of time that the server daemon must wait for the user to login. The server daemon is disconnected after this period of time has elapsed. If the time period is set to 0, then there is no time limit for the user to login. The server wait till the user logs in. The unit of time is set in seconds. The default value for this directive is 120 seconds.

LogLevel

Use this directive to specify the verbosity level that is used when logging messages from the `sshd` daemon. The possible values are:

- `QUIET`
- `FATAL`
- `ERROR`
- `INFO`
- `VERBOSE`
- `DEBUG`
- `DEBUG1`
- `DEBUG2`
- `DEBUG3`

The default value is `INFO`. The `DEBUG` and `DEBUG1` values are equivalent. `DEBUG2` and `DEBUG3` each specify higher levels of debugging output. HP recommends that you do not enable logging with a `DEBUG` level as it violates user privacy.

LogSftp

Use this directive to specify if the sftp-server subsystem transactions must be logged or not. The argument must be `YES` or `NO`. The default value is `NO`.

MACs

This directive specifies the available Message Authentication Code (MAC) algorithms. The MAC algorithm is used in protocol version 2 for data integrity protection. Multiple algorithms must be comma separated. The default setting is `hmac-md5,hmac-sha1,hmac-ripemd160,hmac-sha1-96,hmac-md5-96`.

MaxAuthTries

Use this directive to specify the maximum number of authentication attempts that are permitted per connection. Once the number of failures reaches half the specified value, additional failures are logged. The default value for this directive is 6 authentication attempts permitted per connection.

MaxStartups

Use this directive to specify the maximum number of concurrent unauthenticated connections to the `sshd` daemon. Additional connections are refused until authentication succeeds or the `LoginGraceTime` expires for a connection. The default value for this directive is 10 concurrent unauthenticated connections to the `sshd` daemon.

PasswordAuthentication

Use this directive to specify if a password must be accepted as proof of identity at login. If `KerberosAuthentication` is disabled, the login password is sufficient. However, when `KerberosAuthentication` is also enabled, the Kerberos Server password is accepted as a proof of identity.

PermitEmptyPasswords

Use this directive to specify whether the server allows users to login to accounts with empty password strings. Use this directive when password authentication is allowed. The default setting for this directive is `NO`.

PermitRootLogin

Use this directive to enable users to login as root using `ssh(1)`. You can use this directive in the following methods:

- Disable all root logins.
- Enable all root logins with any authentication method.
- Enable root logins with limited authentication methods.

PermitUserEnvironment

Use this directive to control environment processing by specifying whether `~/.ssh/environment` and `environment= options` in the `~/.ssh/authorized_keys` file are processed by the `sshd` daemon. The `~/.ssh/environment` file must be writable only by the user; it need not be readable by anyone else. Enabling environment processing enables users to bypass access restrictions. This option is automatically disabled when the `UseLogin` option is enabled.

PidFile

Use this directive to specify where to look for the process `id` of the daemon. This file contains the latest instance of the running `sshd` daemon, if multiple `sshd` daemons are running. This file is empty if there is no listening daemon. This directive is ineffective when the server daemon is started in the debug mode.

Port

Use this directive to ensure that the `sshd` daemon listens on a particular port. The client must also connect to the same port on which the daemon is listening. The default value for this directive is 22.

PrintLastLog

Use this directive to display the date and time when the user last logged in. This information is displayed when the user logs in the next time. The default setting for this directive is `YES`.

PrintMotd

Use this directive to display information at login time. When you enable this directive, the `sshd` daemon prints information from the `/etc/motd` file when a user logs in. The default setting for this directive is `YES`.

Protocol

Use this directive to specify the version number of the protocol that the `sshd` daemon supports. You can specify multiple versions of the protocol. Multiple versions of the protocol must be comma separated. This

directive specifies only the list of protocol versions available for the client to select. It does not specify the priority in protocols. For example, specifying 2,1 is identical to specifying 1,2.

PubkeyAuthentication

Use this directive to enable `PublickeyAuthentication`. When this directive is enabled, The sshd daemon uses cryptographic keys to verify users identity. If you are using SSH-protocol version 1 and `PubkeyAuthentication` is enabled and `RSAAuthentication` is set to `NO`, then the next authentication method is looked up. However, if you are using SSH-protocol version 2, the `DSA` authentication method is used.

RhostsRSAAuthentication

Use this directive to perform RSA-based host authentication in addition to standard `.rhosts` or `/etc/hosts.equiv` authentication. This directive does not work for outbound connections from privileged ports. This directive is applicable for SSH-protocol version 1 only.

RSAAuthentication

Use this directive to specify whether pure RSA authentication is allowed. This directive is applicable for SSH-protocol version 1 only.

SftpLogFacility

Use this directive to specify the facility code to be used when logging sftp-server transactions. Following lists the valid values for this directive:

- `DAEMON`
- `USER`
- `AUTH`
- `LOCAL0`
- `LOCAL1`
- `LOCAL2`
- `LOCAL3`

- LOCAL4
- LOCAL5
- LOCAL6
- LOCAL7

The default setting for this directive is AUTH.

SftpLogLevel

Use this directive to specify the verbosity level to be used when logging messages from sftp-server. Following lists the valid values for this directive:

- QUIET
- FATAL
- ERROR
- INFO
- VERBOSE
- DEBUG
- DEBUG1
- DEBUG2
- DEBUG3

The values DEBUG and DEBUG1 are equivalent. DEBUG2 and DEBUG3 each specify higher levels of debugging output. HP recommends that you do not enable logging with a DEBUG level as it violates user privacy. The default setting for this directive is INFO.

SftpPermitChmod

Use this directive to specify if the sftp client can execute chmod commands on the server. The default setting for this directive is YES.

SftpPermitChown

Use this directive to specify whether the sftp client can execute chown or chgrp commands on the server. When this directive is set to YES, the sftp client can execute both chown and chgrp commands. The default setting for this directive is YES.

SftpUmask

Use this directive to specify an optional umask command for sftp-server subsystem transactions. If a umask is specified, this umask overrides all system, environment or sftp client permission modes. If no umask or an invalid umask command is provided, file creation mode defaults to the permission mode specified by the sftp client. The default setting for this directive is no umask.

StrictModes

Use this directive to check access rights and permissions for files. When you enable this directive, the sshd daemon checks the file modes, user file ownership, and home directory before accepting a user login. HP recommends setting this directive to YES because users can sometimes accidentally leave their directories or files world-writable.

SyslogFacility

Use this directive to specify the facility code to be used when logging messages from the sshd daemon. Following lists the valid values for this directive:

- DAEMON
- USER
- AUTH
- LOCAL0
- LOCAL1
- LOCAL2
- LOCAL3
- LOCAL4

- LOCAL5
- LOCAL6
- LOCAL7

The default setting for this directive is AUTH.

TCPKeepAlive

Use this directive to control the flow of TCP KeepAlive messages to other side. If they are sent, the connection is terminated or one of the machines crash, avoiding infinitely hanging sessions.

When you enable this directive, connections are terminated if the route is down temporarily. However, if Keepalive messages are not sent, sessions can hang indefinitely on the server, leaving ghost users and consuming server resources. The default setting for this directive is YES.

UseDNS

Use this directive to specify the order in which the sshd daemon looks up the remote host name and checks that the resolved host name for the remote IP address maps back to the very same IP address. This directive was called VerifyReverseMapping (default no) in previous versions of HP-UX Secure Shell. The default setting for this directive is YES.

UseLogin

Use this directive to specify whether to use login(1) for interactive login sessions. Enabling this option automatically disables X11Forwarding because login(1) does not handle xauth cookies. When you enable UseLogin, the PermitUserEnvironment directive is automatically disabled.

UsePAM

Use this directive to enable PAM authentication and session set-up. HP recommends disabling password authentication when enabling the UsePAM directive. When PasswordAuthentication and UsePAM are set to YES, the password is checked for three times if a password fails to authenticate. A new prompt is then displayed indicating that ssh is failing back to Password authentication.

UsePrivilegeSeparation

Use this directive to specify whether sshd separates privileges by creating an unprivileged child process to deal with incoming network traffic. After successful authentication, another process is created that has the privilege of the authenticated user. By enabling the UsePrivilegeSeparation directive, you can prevent privilege escalation by containing any corruption within the unprivileged processes.

X11DisplayOffset

Use this directive to specify the first display number that the sshd daemon uses for X11 forwarding. This is used to prevent sshd crashing the real X11 servers. The default value for this directive is 10.

X11Forwarding

Use this directive to enable X11 forwarding. Valid arguments are YES or NO. In OpenSSH 4.0, the default setting is NO. In HP-UX Secure Shell, the default setting is modified to YES. When this directive is enabled, there is additional exposure to the server and client displays if the sshd proxy display is configured to listen on the wildcard address (see “X11UseLocalhost” on page 50 for more details). Additionally, authentication spoofing, authentication data verification, and substitution occur on the client side. The security risk of using this directive is that the X11 display server of the client can be exposed to attack when the client requests forwarding.

X11UseLocalhost

Use this directive to bind the X11 forwarding server to the loopback address or the wildcard address. In case of loopback address, the hostname part of the DISPLAY environment variable is localhost. This prevents remote hosts from connecting to the proxy display.

However, some older X11 clients cannot function with this configuration. Set the X11UseLocalhost directive to NO to bind the forwarding server to the wildcard address.

In OpenSSH 4.0, the default setting is set to YES. In HP-UX Secure Shell the default setting is modified to NO.

Default Configuration Directive Settings in the `sshd_config` File

Table 12 compares the default configuration directive settings in the `sshd_config` file in HP-UX Secure Shell A.04.00 in comparison with previous versions of HP-UX Secure Shell.

NOTE A * in front of a value implies that this default on HP-UX Secure Shell is modified from its original OpenSSH default value.

Table 12 Comparative Default Configuration Settings

Configuration Directive	HP-UX Secure Shell Version 3.8	HP-UX Secure Shell Version 3.9	HP-UX Secure Shell Version 4.0
`AcceptEnv`	Not available	No environment variables accepted	No environment variables accepted
AddressFamily	Not available	Not available	any
`AllowGroups`		Login allowed for all groups	Login allowed for all groups
`AllowTcp-Forwarding`	`YES`	`YES`	`YES`
`AuthorizedKeysFile`	`.ssh/authorized_keys`	`.ssh/authorized_keys`	`.ssh/authorized_keys`
`ChallengeResponse-Authentication`	`YES`	`YES`	`YES`
`ClientAliveCount-Max`	`3`	3	`3`
`ClientAliveInterv-al`	0	`0`	`0`
`Compression`	`YES`	YES	YES
`DenyGroups`	Login allowed for all groups	Login allowed for all groups	Login allowed for all groups

Table 12 Comparative Default Configuration Settings (Continued)

Configuration Directive	HP-UX Secure Shell Version 3.8	HP-UX Secure Shell Version 3.9	HP-UX Secure Shell Version 4.0
DenyUsers		Login allowed for all users	Login allowed for all users
GatewayPorts	NO	NO	NO
GSSAPI-Authentication	NO	NO	NO
GSSAPICleanup-Credentials	YES	YES	YES
GSSAPIEnable-MitmAttack	NO	NO	NO
Hostbased-Authentication	NO	NO	NO
HostKey		/opt/ssh/etc/ssh_host_key for protocol version 1 /opt/ssh/etc/ssh_host_rsa_key and /opt/ssh/etc/ssh_host_dsa_key for protocol version 2	/opt/ssh/etc/ssh_host_key for protocol version 1 /opt/ssh/etc/ssh_host_rsa_key and /opt/ssh/etc/ssh_host_dsa_key for protocol version 2
IgnoreRhosts	YES	YES	YES
IgnoreUserKnown-Hosts	NO	NO	NO
Kerberos-Authentication	YES*	YES*	YES*
KerberosOrLocal-Passwd	YES	YES	YES
KerberosTicket-Cleanup	YES	YES	YES

Table 12 Comparative Default Configuration Settings (Continued)

Configuration Directive	HP-UX Secure Shell Version 3.8	HP-UX Secure Shell Version 3.9	HP-UX Secure Shell Version 4.0
ListenAddress	Listen on all local addresses	Listen on all local addresses	Listen on all local addresses
LoginGraceTime	120	120	120
LogLevel	INFO	INFO	INFO
LogSftp	Not available	NO	NO
MACs	hmac-md5,hmac-sha1,hmac-ripemd160,hmac-sha1-96,hmac-md5-96	hmac-md5,hmac-sha1,hmac-ripemd160,hmac-sha1-96,hmac-md5-96	hmac-md5,hmac-sha1,hmac-ripemd160,hmac-sha1-96,hmac-md5-96
MaxAuthTries	Not available	6	6
MaxStartups	10	10	10
Password-Authentication	YES	YES	YES
PermitEmpty-Passwords	NO	NO	NO
PermitRootLogin	YES	YES	YES
PermitUser-Environment	NO	NO	NO
PidFile	/var/run/sshd.pid	/var/run/sshd.pid	/var/run/sshd.pid
Port	22	22	22
PrintLastLog	YES	YES	YES
PrintMotd	YES	YES	YES
Protocol	2	2	2
Pubkey-Authentication	YES	YES	YES

Table 12 Comparative Default Configuration Settings (Continued)

Configuration Directive	HP-UX Secure Shell Version 3.8	HP-UX Secure Shell Version 3.9	HP-UX Secure Shell Version 4.0
RhostsRSA-Authentication	NO	NO	NO
RSAAuthentication	YES	YES	YES
SftpLogFacility	Not available	AUTH	AUTH
SftpLogLevel	Not available	INFO	INFO
SftpPermitChmod	Not available	YES	YES
SftpPermitChown	Not available	YES	YES
SftpUmask	Not available	<no umask>	<no umask>
StrictModes	YES	YES	YES
SyslogFacility	AUTH	AUTH	AUTH
TCPKeepAlive	YES	YES	YES
UseDNS	YES	YES	YES
UseLogin	NO	NO	NO
UsePAM	YES*	YES*	YES*
UsePrivilege-Separation	YES	YES	YES
X11DisplayOffset	10	10	10
X11Forwarding	YES*	YES*	YES*
X11UseLocalhost	NO*	NO*	NO*

HP-UX Secure Shell and chroot environments

HP-UX Secure Shell A.04.00 supports `chroot` functionality for the `ssh`, `sftp`, and `scp` commands. The `chroot` functionality is mainly used as an added security measure.

When you enable `chroot`, you can start an application in a specified directory and allow all its users access to that directory and the directories below it. It prevents users from using the `cd` command to access directories at a higher level. Use this functionality to enable restricted file and directory access to users of a particular application. This is not an end-user feature. The system administrator must enable the `chroot` functionality for an application. All users of that application will automatically be subject to the restrictions imposed by `chroot`.

Refer to the `README` file at `/opt/ssh/README.hp` for more information on setting up the `chroot` functionality. The `chroot` setup script is available at `/opt/ssh/ssh_chroot_setup.sh`.

Frequently Asked Questions (FAQ)

This section discusses questions frequently asked about HP-UX Secure Shell.

What is the difference between HP-UX Secure Shell A.04.00 and OpenSSH 4.0p1?

OpenSSH 4.0p1 is the latest free version of the SSH protocol suite of network connectivity tools. OpenSSH supports SSH protocol versions 1.3, 1.5, and 2.0.

HP-UX Secure Shell is a binary package compiled with support for PAM, `gssapi`, `krb5`, `libwrap`, and no support for Smartcard. You can install and remove HP-UX Secure Shell using the SD-UX utility.

Why must I use Secure Shell?

The standard services for interactive sessions on remote machines are insecure and make the server system vulnerable to attacks. HP-UX Secure Shell offers strong encryption at authentication and for the whole session, which makes it a perfect replacement for these services.

What is the authentication process in HP-UX Secure Shell?

Secure Shell authenticates using one or more of the following authentication mechanisms:

- Password (the `/etc/passwd` or `/etc/shadow` in UNIX)
- User public key (RSA or DSA, depending on the release)
- Kerberos5/GSSAPI for SSH-2
- Host-based authentication
- Challenge-Response authentication (also known as Keyboard-Interactive)

What are the features that HP-UX Secure Shell provides?

HP-UX Secure Shell supports both SSH-1 and SSH-2 protocols. HP recommends that you use SSH-2 protocol to avoid an insertion attack on your server.

Does HP-UX Secure Shell support Smartcard authentication?

No. HP-UX Secure Shell is compiled without Smartcard support.

Does HP support recompiled versions of HP-UX Secure Shell?

HP does not support recompiled versions of HP-UX Secure Shell.

What are the limitations of this product?

Secure Shell is not a true shell unlike the UNIX Bourne shell and C Shell. It does not provide a complete security solution.

Does installing HP-UX Secure Shell product require a kernel rebuild?

No. HP-UX Secure Shell is an application level protocol and does not require a kernel rebuild or system reboot.

How can I uninstall HP-UX Secure Shell?

Use the `swremove` command to remove HP-UX Secure Shell.

How do I find out which version of HP-UX Secure Shell I am using? How do I find out whether I am running HP-UX Secure Shell or the public domain version of OpenSSH?

Use the `swlist` command to display the name and version number of HP-UX Secure Shell. For example:

```
# swlist | grep T1471
T1471AA A.04.00.000 HP-UX Secure Shell
```

You can also use the `what` command shown in the example below:

```
 # what /usr/bin/scp
```

How is the performance of HP-UX Secure Shell?

Compared with conventional file transfer methods, the `scp` command is 2 - 3 times slower than `rcp`, and `sftp` is 2 to 3 times slower than ftp. This is because HP-UX Secure Shell authenticates both the server and users, and encrypts both the data and the password. In addition, HP recommends you use the `/dev/random` device on your system to significantly speed up program initialization.

Do the `rsync` and `rdist` utilities support HP-UX Secure Shell?

You cannot specify `ssh` as the connection mechanism under HP `rdist`. HP does not supply `rsync`. HP has not tested HP-UX Secure Shell with the open source version of `rdist` or `rsync`.

Does HP-UX Secure Shell support the `DenyHosts` parameter?

HP-UX Secure Shell does not support the `AllowHosts`, `DenyHosts`, `DenySHosts`, and `IgnoreRootRhosts` parameters. HP-UX Secure Shell supports the `AllowUsers` and `DenyUsers` parameters.

How can I configure HP-UX Secure Shell to allow multiple users access to an SFTP server using one login using an encrypted connection?

Use public key authentication to allow multiple users access to an SFTP server through an encrypted connection. Each local user gets a public and private key pair. All of the public keys are added to the `~/.ssh/authorized_keys` file of a single user on a remote machine. Each local user can then issue the `sftp` command and log in as the remote user and all local users will share access to the remote user. Note that all of the local users can also use `ssh` to access the remote user.

What diagnostic tools are available with HP-UX Secure Shell? Where can I find error messages, and log files?

HP-UX Secure Shell logs debug and error messages using the `syslog` protocol. Logging is controlled by the `SyslogFacility` and `LogLevel` configuration keywords. Use the appropriate `syslog` log levels (`QUIET`, `FATAL`, `ERROR`, `INFO`, `VERBOSE`, and `DEBUG`) to gather more information about error scenarios. As defined by the `sshd_config` file, the default for `syslogFacility` is set to `AUTH` and `LogLevel` is set to `INFO`.

If `sshd` runs in debug mode (`-d`), logging goes to standard error instead of to `syslog`. You can get more debugging information by using additional d's for `sshd` and more v's for ssh. For example:

```
ssh -v

ssh -vv

ssh -vvv

sshd -d

sshd -dd

sshd -ddd
```

Other commands with debugging option `-v` are:

```
ssh-keyscan -v
sftp -v
scp -v
ssh-keyscan -v
```

Is `libwrap.a` linked in HP-UX Secure Shell? Must I only configure `hosts.allow` and `hosts.deny` to use the access control provided by `tcp_wrapper`?

Yes, the `libwrap.a` archive library consisting of `tcp_wrapper version 7.6`, is linked to HP-UX Secure Shell. You only need to configure `hosts.allow` and `hosts.deny` to use the access control provided by `tcp_wrapper`.

Is HP-UX Secure Shell vulnerable to the reported double free bug in the `zlib` compression algorithm documented at http://www.cert.org/advisories/CA-2002-07.html?

All versions of HP-UX Secure Shell starting from A.03.10 are built with support for `zlib-1.1.4` or later. So, HP-UX Secure Shell is not affected by the bug described above.

HP-UX Secure Shell A.04.00 is built with zlib v1.2.2.

Is HP-UX Secure Shell vulnerable to the following CERTs:

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0147

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0131?

This version of HP-UX Secure Shell is built with OpenSSL-0.9.7e and is not affected by these two CERTs. The vulnerabilities were fixed in OpenSSL-0.9.7d.

What options are HP-UX Secure Shell compiled with?

HP-UX Secure Shell is compiled with the following options defined in the `config.h`, `ssh.h`, and `makefile` files.

Defined in the `config.h` file

- `#define USE_PAM 1`

- `#define IPV4_IN_IPV6 1`
- `#define GSSAPI 1`
- `#define KRB5 1`
- `#define LIBWRAP 1`
- `#define HAVE_MD5_PASSWORDS 1`
- `#undef SMARTCARD`

Defined in the `ssh.h` file:

- `#define SSH_DEFAULT_PORT 22`
- `#define SSH_SERVICE_NAME ssh`

Defined in `makefile`:

- `prefix=/opt/ssh`
- `mandir=/opt/ssh/share/man`
- `piddir=/var/run`
- `PRIVSEP_PATH=/var/empty`
- `bindir=/opt/ssh/bin`
- `sbindir=/opt/ssh/sbin`
- `xauth_path=/usr/bin/X11/xauth`
- `sysconfdir=/opt/ssh/etc`
- `LIBPAM=-lpam`

Cisco routers and switches are enabled with SSH-1 and only use DES. How do I configure HP-UX Secure Shell to work with CISCO SSH-1?

SSH protocol version-1 is disabled in `ssh_config` by default. You can either modify the configuration file or override the protocol on the command line. The client supports Data Encryption Standard (DES), but the server does not support DES. Issue the following command to configure HP-UX Secure Shell to work with CISCO SSH-1:

```
# ssh -1 -c des <command line options like user and machine
name>
```

Can I use a Secure Shell connection to `swinstall` (SD-UX) to a system in a secure way when the two systems are separated by a firewall?

HP-UX Secure Shell handles simple connection tunneling. SD-UX uses more than one connection. SD-UX checks to see the system it is running on and the system you are trying to connect to. It can use UDP, which HP-UX Secure Shell does not support. Use a depot file to secure the communication. Use `swpackage` to create a depot file. Use `sftp` or `scp` to copy the depot file to the local machine and then use `swinstall` to install the depot file. This procedure ensures that the network traffic is secure. However, you must get the correct depot file yourself, instead of letting SD choose one that is appropriate for your OS.

What is `chroot`? How does it work? How does it get set up, and what does the user have to do to make use of it? Where is chroot supported in Secure Shell?

HP-UX Secure Shell A.04.00 supports `chroot` functionality for the `ssh`, `sftp`, and `scp` commands. The `chroot` functionality is mainly used as an added security measure.

The `chroot` functionality allows an application to start in a specified directory and allows all its users from accessing that directory and the directories below it. It restricts the user from doing a `cd` command above that specified directory. Use this functionality to enable restricted file and directory access to users of a particular application. The `chroot` functionality is not an end-user feature. The system administrator must enable `chroot` for an application. All users of that application will automatically be subject to the restrictions imposed by `chroot`

For more information on setting up the `chroot` functionality, refer the `README` file at `/opt/ssh/README.hp`. The `chroot` setup script is available at `/opt/ssh/ssh_chroot_setup.sh`.