HP-UX Secure Shell A.04.20.004/005 Release Notes

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

Manufacturing Part Number: T1471-90019

December 2005

U.S.A

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

HP-UX Secure Shell A.04.20.004/005

HP-UX Secure Shell A.04.20.004/005

This chapter discusses the new features in HP-UX Secure Shell A.04.20.004/005 (A.04.20).

Secure Shell Versions on HP-UX

Table 1 lists the versions of HP-UX Secure Shell products available for HP-UX 11.0, 11i v1, and 11i v2.

Table 1 **Secure Shell on HP-UX**

Version	Supported Operating System
HP-UX Secure Shell Version A.04.20.004	HP-UX 11.0 HP-UX 11i v1
HP-UX Secure Shell Version A.04.20.005	HP-UX 11i v2

New Features

The following new features have been introduced in HP-UX Secure Shell A.04.20 as compared to the previous release of HP-UX Secure Shell (A.04.10):

- “High Performance Enabled SSH/SCP Patch” on page 7
- “Configuration Directives in the Server” on page 8
  - “The CountKeyAuthBadLogins Directive” on page 8
  - “The EnforceSecureTTY Directive” on page 9
- “Auth Selection Patch” on page 16
- “Increase in the Default Size of RSA and DSA Keys” on page 18
- “Delayed Compression” on page 18
- “Support for Improved Arcfour Cipher Modes” on page 19
- “Modified ControlPath Client Configuration Directive” on page 19
- “Support for X11 and Agent Forwarding Over Multiplexed Connections” on page 20

The subsequent sections contain detailed descriptions of the new features.

High Performance Enabled SSH/SCP Patch

A new High Performance Enabled SSH/SCP (HPN) patch that enables Secure Shell to take advantage of high speed networks. has been included in this version of HP-UX Secure Shell. This patch enables applications to take advantage of the large `tcp` send and receive buffers that are available in high bandwidth interfaces.

The HPN patch consists of the following internal changes:

- A `-w` command-line option is added to the ssh client. This option enables the user to specify the kernel `tcp` send and receive buffer size without affecting the buffer sizes used by other applications on the system.

- The HPN patch optimizes the client-server window size. The client-server window is the amount of information that the ssh client writes to the tcp socket before waiting to receive an ack packet from the server. Without the HPN patch, the window size is set at 128 KB. With this patch, the window size is adjusted based on the tcp buffer size. The window used by the SSH client-server communication is set internally and can be upto a maximum of 2**29 - 1 bytes.
- The buffer used by the scp client for reading files is increased from 2 KB to 16 KB.

NOTE A client with the HPN patch can work with a non-HPN-enabled sshd server. However, the client-server connection does not result in optimal performance. While establishing a connection, the ssh client checks the server version. If the server version has a -hpn suffix, the client assumes that the server is HPN-enabled, and dynamically adjusts the window size based on the tcp buffer size. If the server version does not have a -hpn suffix, the client assumes that the server does not have the HPN patch, and retains the window size to 128 KB.

A properly tuned TCP/IP stack is a prerequisite to obtain optimal results with the HPN patch.

For more information, refer to the white paper at http://www.psc.edu/networking/projects/hpn-ssh/.

Configuration Directives in the Server

The following server configuration directives have been added in this version of HP-UX Secure Shell:

- “The CountKeyAuthBadLogins Directive” on page 8
- “The EnforceSecureTTY Directive” on page 9

The CountKeyAuthBadLogins Directive

An enhancement (enforcement of maximum bad login attempts) provided in a previous version of HP-UX Secure Shell resulted in several btmp records. Due to this enhancement, the real btmp records cannot be distinguished from the btmp records that are written because a key-based authentication method is expected to fail.

The default value of the CountKeyAuthBadLogins configuration directive is set to NO by default. This means that failed login attempts for key-based authentication methods do not generate btmp logs, and they do not count towards bad login attempts.

This configuration directive is not a part of the OpenSSH 4.2 base code. It is an enhancement provided in HP-UX Secure Shell only.

The EnforceSecureTTY Directive

HP-UX Secure Shell reads the tty (terminal type) names that are listed in the /etc/securetty file. HP-UX Secure Shell then restricts Secure Shell root logins to only those ttys that are listed in the etc/securetty file. Any root login attempt from a tty that is not listed in the /etc/securetty file is rejected with a login failure error.

In this version of HP-UX Secure Shell, you can configure the tty restriction using the EnforceSecureTTY configuration directive. The default value of the EnforceSecureTTY configuration directive is NO. By default, the tty restriction is not enforced.

You can use the EnforceSecureTTY configuration directive in conjunction with the PermitRootLogin configuration directive. Table 2 describes the behavior of the ssh, scp, and sftp commands with different combinations of the EnforceSecureTTY and PermitRootLogin configuration directives.

Table 2 Behavior of the ssh, scp, and sftp Commands With Different Combinations of EnforceSecureTTY and PermitRootLogin

Enforce Secure TTY	PermitRoot Login	Behavior of the ssh Command	Behavior of the scp and sftp Commands
NO	NO	Host login[a] and hostcommand[b] execution is disallowed for all root users	Root users cannot execute the scp and sftp[c] commands regardless of /etc/securetty.
NO	YES	Host login and hostcommand execution is allowed for all root users.	Root users can execute the scp and sftp commands regardless of /etc/securetty.

Table 2 **Behavior of the ssh, scp, and sftp Commands With Different Combinations of EnforceSecureTTY and PermitRootLogin**

Enforce Secure TTY	PermitRoot Login	Behavior of the ssh Command	Behavior of the scp and sftp Commands
YES	NO	Host login and hostcommand execution is disallowed for all root users.	Root users cannot execute the `scp` and `sftp` commands regardless of `/etc/securetty.`
YES	YES	Host login is allowed only for those root users whose `ptys` are listed in the `/etc/securetty` file. Hostcommand execution is allowed for all root users (independent of `/etc/securetty`).	Root users can execute the `scp` and `sftp` commands regardless of `etc/securetty`.

Table 2 Behavior of the ssh, scp, and sftp Commands With Different Combinations of EnforceSecureTTY and PermitRootLogin

Enforce Secure TTY	PermitRoot Login	Behavior of the ssh Command	Behavior of the scp and sftp Commands
`YES`	`Forced-Commands Only`	Host login and hostcommand execution is disallowed for all root users independent of `/etc/securetty`. Forced-command[d] execution is dictated by the `pty` or `no-pty` option. This option is specified in the `authorized_keys` file, located in the home directory of the root user, on the server. The default option is `pty`. If run with a `pty` option, then forced_command execution is allowed only for root users whose `ptys` are listed in the `/etc/securetty` file. If run with a `no-pty` option, then forced-command execution is allowed for all root users independent of `/etc/securetty`. **NOTE:** For Forced-Commands-only, root users must log in using the PublicKey authentication method, but this additional requirement is unrelated to the `EnforceSecureTTY` enhancement. This applies to the `ssh`, `scp`, and `sftp` commands.	Forced-command execution is allowed for all root users independent of `/etc/securetty`, and independent of the `pty` setting in the `authorized_keys` file. However, no `pty` will be allocated (even if specified in the `authorized_keys` file). **IMPORTANT:** The `scp` and `sftp` commands, and forced_command are mutually exclusive. If forced_command execution is set, only the "forced command" is executed, and no file transfers are allowed.

Table 2 Behavior of the ssh, scp, and sftp Commands With Different Combinations of EnforceSecureTTY and PermitRootLogin

Enforce Secure TTY	PermitRoot Login	Behavior of the ssh Command	Behavior of the scp and sftp Commands
`YES`	`Without Password`	Host login is allowed only for root users whose `ptys` are listed in the `/etc/securetty` file (these root users must log in with a method other than "password", but this additional requirement is unrelated to the `EnforceSecureTTY` enhancement). Hostcommand execution is allowed for all users regardless of `/etc/securetty`.	Root users can execute the `scp` and `sftp` commands independent of `/etc/securetty` (these root users must log in with a method other than "password").

Table 2 **Behavior of the ssh, scp, and sftp Commands With Different Combinations of EnforceSecureTTY and PermitRootLogin**

Enforce Secure TTY	PermitRoot Login	Behavior of the ssh Command	Behavior of the scp and sftp Commands
`NO`	`Forced-Commands-Only`	Host login and hostcommand execution is disallowed for all root users. Forced-Commands execution is allowed for all root users. **NOTE:** For Forced-Commands-only, root users must log in using the PublicKey authentication method, but this additional requirement is unrelated to the `EnforceSecureTTY` enhancement. This applies to the `ssh`, `scp`, and `sftp` commands.	Forced-command execution is allowed for all root users independent of `/etc/securetty` and the `pty` setting in the `authorized_keys` file. However, no pty will be allocated (even if specified in the `authorized_keys` file). **IMPORTANT:** The `scp` and `sftp` commands, and forced_command are mutually exclusive. If forced_command execution is set, only the "forced command" is executed, and no file transfers are allowed.
`NO`	`Without Password`	Host login is allowed for all root users (these root users must log in with a method other than "password", but this additional requirement is unrelated to the `EnforceSecureTTY` enhancement). Hostcommand execution is allowed for all users regardless of `/etc/securetty`.	Root users can execute the `scp` and `sftp` commands regardless of `etc/securetty` (these root users must log in with a method other than “password”).

a. Host login refers to a client directly logging into a host. Following is an example of Host login:

```
$ ssh hostxyz
```

b. Hostcommand execution refers to a client executing only one command against a server. The client logs into the server, executes the command, and exits. Following is an example of the Hostcommand execution:

```
$ ssh hostxyz ls /tmp
```

c. The execution of the `scp` and sftp commands is similar to that of the `hostcommand`. However, no `pty` is allocated for `scp` and `sftp`, and the `/etc/securetty` file is not checked. Any combination of `EnforceSecureTTY` and `PermitRootLogin` that allows hostcommand execution for `ssh` also allows `scp` and `sftp` execution.

d. Forced-command execution refers to a client executing a command predefined in the `authorized_keys` file of the client. This file is located in the home directory of the client, on the server.

The `EnforceSecureTTY` directive has been implemented to work with `UseLogin YES` as well as `UseLogin NO`. Although the `login(1)` function already has the code to check `/etc/securetty`, that code is part of authentication. When `UseLogin` is set to `YES`, Secure Shell invokes `login(1)` with the `do not authenticate` flag. As a result, the `/etc/securetty` part of the `login(1)` code is skipped.

Therefore, it becomes necessary for Secure Shell to read and process `/etc/securetty` even when `UseLogin` is set to `YES`, and that is how this patch has been implemented.

NOTE

Users accustomed to how `telnet` behaves with `/etc/securetty` will find one difference between `telnet` and this new release of Secure Shell. In telnet, a `pty` is allocated to the user connection before authentication. In Secure Shell, authentication must succeed in order for `sshd` to do pty allocation. Once authentication succeeds, `sshd` does not come back and re-prompt the user for a password.

Table 3 describes the behavioral difference between telnet and Secure Shell logins.

Table 3 **Difference in Behavior Between telnet and ssh Logins**

A telnet Login	An SSH Login
When a root user attempts a telnet login with a tty that is not listed in /etc/securetty), telnet continues to prompt the user for a password regardless of whether the user types a valid or invalid password.	If the EnforceSecureTTY configuration directive is set to YES, and a root user attempts an ssh login with a tty not listed in the /etc/securetty, HP-UX Secure Shell continues to prompt the user for a password as long as the user enters invalid passwords. Once the user enters a valid password, the sshd daemon does the following: • Authenticates the user • Allocates a pty (psuedoterminal or pseudoteletype) • Finds out that the pty is not permitted • Closes the connection

The change in the behavior of the ssh login is due to a fundamental difference in the behavior between telnet and ssh logins. In telnet, a pty is allocated before authentication. In HP-UX Secure Shell, authentication must succeed before a pty is allocated. Once authentication succeeds, HP-UX Secure Shell does not re-prompt the user for passwords.

NOTE Users accustomed to remsh will also find one point of difference with the way remsh handles securetty. For both Secure Shell and remsh, the forced_command only option will be allowed even for root users coming in from a tty not listed in /etc/securetty. The difference between the Secure Shell implementation and remsh behavior is that for remsh, /etc/securetty is enforced only for the password authentication method. If a remsh user authenticates through host-based authentication, remsh will not enforce /etc/securetty. In the Secure Shell implementation, there is no distinction between authentication methods with regard to /etc/securetty.

Auth Selection Patch

A new 3rd-party patch called the Auth Selection patch has been included in this version of HP-UX Secure Shell. This patch enables the sshd daemon to restrict individual users to specific authentication methods. 12 new configuration directives have been added to implement this feature. These configuration directives can be broadly classified as Allow and Deny configuration directives. Table 4 lists the 12 new configuration directives.

Table 4 **New Configuration Directives**

Allow Configuration Directives	Deny Configuration Directives
KerberosAuthAllowUsers	KerberosAuthDenyUsers
KerberosOrLocalPasswdAuthAllowUsers	KerberosOrLocalPasswdAuthDenyUsers
PubkeyAuthAllowUsers	PubkeyAuthDenyUsers
HostbasedAuthAllowUsers	HostbasedAuthDenyUsers
ChallRespAuthAllowUsers	ChallRespAuthDenyUsers
PasswordAuthAllowUsers	PasswordAuthDenyUsers

These configuration directives are similar to the AllowUsers and DenyUsers configuration directives. However, these new configuration directives enable or deny users for that specific authentication method. The * wildcard represents all users. For example,

```
PubKeyAuthAllowUsers *
```

enables all users to log in using public-key authentication (provided users have set up the appropriate the key pairs).

By default, all the Allow configuration directives enable all users, and all the Deny configuration directives deny no users. All these configuration directives have been specified in the sshd_config file that comes with the product, and have been set to their respective default value.

The following scenarios depict how the `sshd` daemon uses the Allow and Deny configuration directives to find out if user U1 is permitted to authenticate using password authentication:

Scenario 1

1. The `sshd` daemon checks if the `PasswordAuthDenyUsers` configuration directive is specified in the `sshd_config` file.
2. If the `PasswordAuthDenyUsers` configuration directive is not specified, the `sshd` daemon checks if the `PasswordAuthAllowUsers` configuration directive is specified.
3. If the `PasswordAuthAllowUsers` configuration directive is not specified, then user U1 is permitted to continue with password authentication.

Scenario 2

1. The `sshd` daemon checks if the `PasswordAuthDenyUsers` configuration directive is specified in the `sshd_config` file.
2. If the `PasswordAuthDenyUsers` configuration directive is specified, then the `sshd` daemon checks to see if user U1 is on the list.
3. If user U1 is on the list, then user U1 is denied password authentication.
4. If user U1 is not on the list, then the `sshd` daemon checks if the `PasswordAuthAllowUsers` configuration directive is specified.
5. If the `PasswordAuthAllowUsers` configuration directive is specified, the `sshd` daemon checks if user U1 is on the list.
6. If user U1 is on the list, then user U1 is permitted to continue with Password authentication.
7. If user U1 is not on the list, then user U1 is denied password authentication.

Scenario 3

1. The `sshd` daemon checks if the `PasswordAuthDenyUsers` configuration directive is specified in the `sshd_config` file.
2. If the `PasswordAuthDenyUsers` configuration directive is specified, then the `sshd` daemon checks to see if user U1 is on the list.

3. If user U1 is on the list, then user U1 is denied password authentication.
4. If user U1 is not on the list, then the sshd daemon checks if the PasswordAuthAllowUsers configuration directive is specified.
5. If the PasswordAuthAllowUsers configuration directive is not specified, then user U1 is permitted to continue with password authentication.

If you want user U1 to use password authentication, you must set the PasswordAuthAllowUsers configuration directive as follows:

```
PasswordAuthAllowUsers U1
```

You do not have to set the PasswordAuthDenyUsers configuration directive. You can use the configuration directive that has fewer members. For example, if you want to enable password authentication for all users other than user U1, you can set the PasswordAuthDenyUsers configuration directive as:

```
PasswordAuthDenyUsers U1
```

For more information on the Auth Selection patch, refer to the http://www.sweb.cz/v_t_m/ webpage.

Increase in the Default Size of RSA and DSA Keys

The default ssh-keygen key length has been increased for RSA and DSA keys from 1024 to 2048 bits. This feature augments the security of key-based authentication methods.

Delayed Compression

In previous releases of HP-UX Secure Shell, the sshd server sends compressed data before the user is successfully authenticated.

In HP-UX Secure Shell A.4.20, the default compression method of the server is changed to compression=delayed. The server invokes the zlib compression modules only after the user is successfully authenticated. This feature eliminates the risk of any zlib vulnerability leading to the server being compromised by unauthenticated users.

IMPORTANT Delayed compression is not supported in older clients. HP-UX Secure Shell releases prior to HP-UX Secure Shell 3.5 will not be able to connect to a new server unless compression is disabled (on the client-side), or the original compression method is enabled on the server (by setting `Compression yes` in the `sshd_config` file).

Support for Improved Arcfour Cipher Modes

Arcfour (RC4) is a stream-based symmetric key algorithm used to encrypt or decrypt data streams. In this version of HP-UX Secure Shell, the first 1536 bytes of a key stream are discarded to ensure that the cipher's internal state is mixed. This feature reduces potential security vulnerabilities with the Arcfour cipher.

NOTE HP does not recommend the Arcfour cipher for high-volume password authentication connections.

Modified ControlPath Client Configuration Directive

The behavior of the `ControlPath` client configuration directive has been modified by adding the following expansion sequences:

- `%h` (target hostname)
- `%p` (target port)
- `%r` (remote username)

A new option, `ControlPath=none` option has been added to disable connection multiplexing.

Refer to the *ssh_config* manpage for more information on the `ControlPath` configuration directive settings.

Support for X11 and Agent Forwarding Over Multiplexed Connections

X11 and agent forwarding is supported over multiplexed connections. The master connection can forward both X11 and agents. The slave connections inherits the `DISPLAY` and `SSH_AUTH_SOCK` environment variables. Even if slaves forward their own agents or X11 details, the server ignores them.

Unsupported Features

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

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

  This configuration directive specifies whether or not to accept forwarded Andrew File System (AFS) tokens.

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

Defects Fixed in HP-UX Secure Shell A.04.20

All defect fixes included in the previous versions of HP-UX Secure Shell are also included in HP-UX Secure Shell A.04.20. Additionally, all defect fixes in OpenSSH 4.2p1 are also included in HP-UX Secure Shell A.04.20. For more information on these fixes, refer to http://bugzilla.mindrot.org

Two security vulnerabilities in OpenSSH versions prior to OpenSSH 4.2p1 have been fixed in this release. Visit the following websites for more information on these security vulnerabilities:

- http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2005-2797
- http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2005-2798

Information on these defect fixes and vulnerabilities is also available at `/opt/ssh/README.hp` on installing HP-UX Secure Shell A.04.20.

Known Problems and Workarounds

Following lists the known problems and workarounds in HP-UX Secure Shell A.04.020:

- Secure Shell user authentication through the public-key will fail in a server environment if the `UsePAM` is set to `YES` and `pam.conf` is set to `PAM_LDAP`.

  Workaround: Currently, there is no workaround for this problem.

- On some systems, when a user logs out of a Secure Shell session, the following message appears in the `syslog.log` file:

  ```
  pam_setcred: error Authentication failed
  ```

  This error message appears only when the daemon is running in the debug mode. This error message is not relevant to (and does not affect) Secure Shell operations. The PAM function `pam_setcred()` generates this message. For root users, it occurs unconditionally. For non-root users, it occurs only when `/usr/sbin/keyserv` is not running on the server system. In a normal `syslogd` operation, the error message does not appear.

- A corner case exists where a Kerberos ticket on a Secure Shell server system can get inadvertently deleted. The following scenario creates this problem:

  1. User U1 creates a Kerberos ticket file on a Secure Shell server system, S1.
  2. The SSH server on S1 is set up for PAM_KERBEROS authentication.
  3. User U1 now remotely connects to the SSH instance on S1 using public-key authentication.
  4. User U1 exits.

  The kinit-generated ticket file created in Step 1 gets deleted when the user exits the Secure Shell session.

  Workaround: Create the Kerberos ticket file (Step 1) in a non-default location, and selectively communicate this file name to Secure Shell processes using the `KRB5CCNAME` environment variable.

- The chroot functionality does not work if the `UseLogin` configuration directive in `sshd_config` is set to `YES`.
- In a chroot-ed environment, users do not see a subset of syslog messages. HP-UX Secure Shell writes syslog messages at the time of authentication and when the session is terminated. The `syslogd` daemon reads the syslog messages written by all subsystems and reports it to the `/dev/log` file. In a chroot-ed environment, the sshd daemon writes its syslog messages to `<newroot>/dev/log`. It is not possible to link the `<newroot>/dev/log` file to the `/dev/log` file, resulting in users not being able to view the subset of syslog messages.

  Workaround: There is no workaround for this problem. Users of chroot-ed HP-UX Secure Shell environments must be aware that a subset of messages written by the sshd daemon will not show up in syslog.

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`. If you are using HP-UX Secure Shell on HP-UX 11i v1, HP recommends that you 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 about 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.*

Prerequisites

This section details the prerequisites for installing HP-UX Secure Shell A.04.20.

Patch Requirements

HP has tested HP-UX Secure Shell A.04.20 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 v1 customers must install these Support Plus patches.

Table 5 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 v1	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 (`http://www.itrc.hp.com`). 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. 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 use 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.04.20:

Table 6 libc Patches

Operating System Version	Patch
HP-UX 11.0	PHCO_25976
HP-UX 11i v1	PHCO_27740

HP recommends that the following PAM patches be installed for use with HP-UX Secure Shell A.04.20:

Table 7 PAM Patches

Operating System Version	Patch
HP-UX 11.0	PHCO_29249
HP-UX 11i v1	PHCO_33215

NOTE The PHCO_33215 patch fixes a PAM-related issue. Without this patch, `pam_acct_mgmt ()` returned success messages on locked accounts. With this patch, account management fails for locked accounts (this is the appropriate behavior). In order to log in using ssh, users must unlock their accounts.

HP recommends that the following pthreads patches be installed for use with HP-UX Secure Shell A.04.20:

Table 8 pthreads Patches

Operating System Version	Patch
HP-UX 11.0	PHCO_26960
HP-UX 11i v1	PHCO_26466

System Requirements

Following are the minimum system requirements for installing HP-UX Secure Shell A.04.20:

Operating System

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

Hardware

- HP/9000 Servers and Workstations
- HP Integrity Servers

Disk Space

Approximately 32MB of disk space

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 v2 Operating Environment (OE)

Software Availability in Native Languages

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

Installing HP-UX Secure Shell

You do not need to remove any previous versions of HP-UX Secure Shell before upgrading to HP-UX Secure Shell A.04.20. However, if you are reverting to an older version of HP-UX Secure Shell, HP recommends that you remove the new product before reverting to 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>` at the command prompt.

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 it is started after installation.

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

HP-UX Secure Shell and chroot environments

HP-UX Secure Shell A.04.20 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 enable 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.04.20 and OpenSSH 4.2p1?

OpenSSH 4.2p1 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.

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.04.20.004 HP-UX Secure Shell
```

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

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

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.04.20 is built with zlib v1.2.3.

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-0.9.7i and is not affected by these two CERTs. The vulnerabilities were fixed in OpenSSL-0.9.7d.