Release Notes for SSH Secure Shell 3.2 pack 6

Contents

1 About This Update Release
- 1.1 Purpose
- 1.2 Version Number Notation

2 CD Contents
- 2.1 AIX Binaries
- 2.2 HP-UX Binaries
- 2.3 Solaris Binaries
- 2.4 Linux Binaries
- 2.5 Windows Binaries
- 2.6 Source Code
- 2.7 Documentation

3 Unix
- 3.1 New Features
- 3.2 Bug Fixes
- 3.3 Known Issues

4 Windows Client
- 4.1 New Features
- 4.2 Bug Fixes
- 4.3 Known Issues

1 About This Update Release

1.1 Purpose

This update release fixes the following issues in the Windows client binaries:

- silent installation did not work.
- X11 tunneling did not work.
- SSH1 usage caused the client to crash.

Other platforms are not affected.

To fix these problems we recommend the upgrade to Windows client version 3.2.9.2. The updated Windows client binaries are version 3.2, patch level 9.2.

1.2 Version Number Notation

“3.2 pack 6” refers to the sixth revision of the 3.2 release CD. Each release CD contains the latest versions of the installation packages for each platform.

“3.2 patch level 9.2” and “3.2.9.2” mean the same thing, and they specify the version of an installation package. Each platform installation package may be of a different patch level.

2 CD Contents

2.1 AIX Binaries

The latest binaries for AIX are version 3.2, patch level 9.

install/aix/ssh-3.2.9-binary-aix-4.3.x.tar.gz
Installation package for AIX 4.3.

install/aix/ssh-3.2.9-binary-aix-5.x.x.tar.gz
Installation package for 5L.

install/aix/README
Installation guide for AIX.

install/aix/license_ssh2.dat
The license file for SSH Secure Shell. Place this in /etc/ssh2 to enable the installed AIX binaries.

2.2 HP-UX Binaries

The latest binaries for HP-UX are version 3.2, patch level 9.

install/hp-ux/ssh-3.2.9-sd-11.00.depot.gz
Installation package for HP-UX 11.x.

install/hp-ux/ssh-3.2.9-sd-10.20.depot.gz
Installation package for HP-UX 10.20.

install/hp-ux/README
Installation guide for HP-UX.

install/hp-ux/license_ssh2.dat
The license file for SSH Secure Shell. Place this in /etc/ssh2 to enable the installed HP-UX binaries.

2.3 Solaris Binaries

The latest binaries for Solaris are version 3.2, patch level 9.

install/solaris/ssh-3.2.9-binary-solaris2.6-7-8-sparc.tar.Z
Installation package for Solaris 2.6, 7 and 8 on Sparc architecture.

install/solaris/README-2.6-7-8
Installation guide for Solaris 2.6, 7 and 8.

install/solaris/ssh-3.2.9-binary-solaris2.5.1-sparc.tar.Z
Installation package for Solaris 2.5.1 on Sparc architecture.

install/solaris/README-2.5.1
Installation guide for Solaris 2.5.1.

install/solaris/license_ssh2.dat
The license file for SSH Secure Shell. Place this in /etc/ssh2 to enable the installed Solaris binaries.

2.4 Linux Binaries

The latest binaries for Linux are version 3.2, patch level 9.

install/linux/ssh-3.2.9-1.i386.rpm
Installation package for Linux.

install/linux/ssh-3.2.9-1.src.rpm
Installation package for Linux.
Included only in the *SSH Secure Shell for Servers* product.

install/linux/README
Installation guide for Linux.

install/linux/license_ssh2.dat
The license file for SSH Secure Shell. Place this in /etc/ssh2 to enable the installed Linux binaries.

2.5 Windows Binaries

The Windows binaries are included only in the *SSH Secure Shell for Workstations* product. The latest binaries for Windows are version 3.2, patch level 9.2.

install/windows/SSHSecureShellClient.exe
SSH Secure Shell Windows client install package.

install/windows/silent-setup.txt
Instructions on how to create a silent installation of the *SSHSecureShellClient.exe*. Using it, you can build a custom installation over a network.

install/windows/silent-setup-client.iss
An InstallShield template for a silent setup of the *SSHSecureShellClient.exe*.

install/windows/accession_commercial_license.txt
SSH Accession commercial license agreement.

2.6 Source Code

install/source/ssh-3.2.9-commercial.tar.gz
SSH Secure Shell for Servers source code. Included only in the *SSH Secure Shell for Servers* product.

2.7 Documentation

index.html
The CD contents front page.

license.html
license.txt
The license agreement in HTML and text formats.

releasenotes.txt, releasenotes.pdf, releasenotes.rtf
This file.

doc/Ssh2winclient-usermanual.pdf
doc/Ssh2winclient-usermanual_html/index.html

The SSH Secure Shell Windows Client User Manual in PDF and HTML formats.

doc/Ssh2unix-quickstart.pdf
doc/Ssh2unix-quickstart_html/index.html

SSH Secure Shell Quick Start Guide in PDF and HTML formats.

doc/Ssh2unix-adminguide.pdf
doc/Ssh2unix-adminguide_html/index.html

SSH Secure Shell Administrator's Guide in PDF and HTML formats.

doc/drafts
SecSh IETF drafts that specify the communications protocols of SSH Secure Shell.

3 Unix

3.1 New Features

In 3.2.9:

- None.

In 3.2.5:

- None.

In 3.2.3:

- scp2: You can now specify the newline convention when using the "-a" option. See manual page scp2(1).
- scp2: Implemented --overwrite, which controls whether to overwrite the destination file(s). Default is "yes", i.e. to overwrite.
- scp2: Implemented interactive mode, i.e. you can make scp2 prompt you whether to overwrite an existing destination file. You can do this by specifying --interactive (-I) on the command line.
- sftp2: Added option "S" and "r" to "ls" (for sorting by size and reversing the sort order, respectively).
- sftp2, scp2: Extensive rewrite of SshFileCopy, and as a consequence, of both scp2 and sftp2 core functionality.

In 3.2.2:

- None.

In 3.2.1:

- None.

In 3.2.0:

sshd2 server

- Added support for certificate external mapper (keyword ExternalMapper).
- Implemented ExternalAuthorizationProgram and PasswdPath configuration options. See sshd2_config(5).
- Implemented ForwardACL configuration parameter for more fine-grained access control for forwards. See sshd2_config(5).
- Implemented AuthPublicKey.{Max,Min}Size, which can specify the size of a public key that may be used to log in. For example, you might specify AuthPublicKey.MinSize to be 2048, after which smaller public keys would not be allowed to log in. Both checks are disabled by default.

  You can also specify the minimum and maximum length for a public key in a certificate with

AuthPublicKey.Cert.{Max,Min}Size. Certificates are available only in commercial and binary versions.

If you want to specify a minimum for both normal public keys and public keys in certificates, you have to use both AuthPublicKey.MinSize and AuthPublicKey.Cert.MinSize options.

- Implemented subconfiguration files. You can specify subconfiguration files with HostSpecificConfig (read in when a client connects) and UserSpecificConfig (read in when the user name the client is trying to log in is known). You can specify multiple subconfiguration files, which will be read in order. Because the subconfiguration files are read after the main daemon has forked, you do not need to restart sshd2 when you modify them (you do need to restart the daemon when you add the keywords describing them to sshd2_config, the main file). The subconfiguration files can have stanzas, or configuration blocks, in them, so you can choose whether you want to use one big subconfiguration file for each {Host,User}SpecificConfig, or whether you want to make smaller configuration files, grouped by the user's group or subnet mask. With this option you can have, for example, different ciphers for different hosts, or you could display different banner messages based on what subnet the client is coming from, or you can modify what authentication methods are required or allowed, based on user and/or group or host the client is logging in to or from.

  You can read more from sshd2_config(5) ({Host,User}SpecificConfig keywords) and sshd2_subconfig(5), which has detailed explanation on the semantics and usage.
- Implemented ResolveClientHostName, with which you can control whether the server will try to resolve the client IP or not. Note that if you set this to "no", you should not set RequireReverseMapping to "yes".
- Added submethod "password" to "keyboard-interactive". This behaves mostly the same way as the standard "password" authentication method. This is not of much use, if you can use PAM (and the "pam" submethod).
- Added submethod "securid" to "keyboard-interactive". This should be used instead of legacy "securid-1@ssh.com".
- Added submethod "pam" to "keyboard-interactive". This method should be used instead of legacy "pam-1@ssh.com".
- PAM authentication no longer needs ssh-pam-client.
- Changed chroot() logic. Now secondary groups will be initialised after chrooting, without the need to copy /etc/group to the chroot jail.
- Added configuration option "ProtocolVersionString" to enable configuration of the (surprise) protocol version string (after the "SSH-{2.0,1.99}-" bit).
- Changed default of Ssh1Compatibility from <set by configure> to "no". So you have to manually set ssh(d)2 to use ssh(d)1, even if ssh(d)1 is installed. (ssh2 has built-in emulation for the SSH1-protocol). This pertains to the ssh2 client as well as the sshd2 server.
- Added support for external hostkeys (keywords HostKeyEkProvider and HostKeyEkInitString). This pertains to the ssh2 client as well as the sshd2 server.
- Added support for Solaris BSM (Basic Security Module).

- Headings are no longer recognized in sshd2_config (as they never should have). Remove the last of the ".*:" and "*:" from your sshd2_config, as no variables after them will be parsed. If sshd2 encounters a heading while parsing the configuration file, it will blurt a big warning message.

ssh2 client

- Integrated the X11 SECURITY extension. If the extension is found, ssh2 informs the Xserver that the client applications should be treated as untrusted by default. If you specify the "+X" command-line option, the X11 clients are treated as trusted, which is essentially the same behaviour as before.
- An exception: If the SECURITY extension is present but we fail to obtain a new cookie via the SECURITY extension, X11 forwarding is disabled. Failing to obtain a cookie via the SECURITY extension is usually a restriction by the Xserver security policy and should be honored by ssh code.

  If this feature causes you problems, you can disable it by configuring with "--without-x11-security". Additional details are under the "TrustX11Applications" option in ssh2_config(5).

  Note that pre-compiled binaries do not support the SECURITY extension, as it requires the X11 shared libraries.
- Implemented support for password changing in password authentication for servers, which support it (mainly Windows servers).
- Implemented SOCKS5 support. To use SOCKS5 when connecting with ssh2, set "UseSOCKS5" to "yes". The default behaviour is to use the old SOCKS4. The client has been changed to use our SOCKS library so that it can now serve both SOCKS4 and SOCKS5 client applications.
- Sanitized exit values a bit. See man ssh2(1).

sftp2 client

- Implemented the "-v" (verbose, i.e. debug level 2) option.
- scp2, sftp2: The destination file is not removed anymore. Instead, if the preserve flag is set, the permissions are set on the file before anything is transferred. Also, if the destination file is not a regular file, do not change its attributes (it is better to err on the side of safety). Otherwise, if root user copied an executable to e.g. /dev/null, the attributes of that device would change.

scp2 client

- Local filenames are not interpreted as regular expressions (regex) anymore on UNIX-like platforms. The behaviour is unchanged for Windows because the "shell" does not understand file globbing.

Other binaries

- sftp-server2: Logging has been added to the beginning and termination of each SFTP-session.

- sshd-check-conf: Updated to understand subconfigurations, and added a command, “dump”, with which you can dump most of the configuration after telling who is logging in and from where.
- ssh-agent2: Fixed bug in ssh-agent2 in which if the user is root and the parent process dies, the process did not die.
- ssh-keygen2: Changed default generated key size to 2048.

Generic Unix changes

- Implemented AuthKbdInt.Plugin configuration option and the “plugin” submethod for “keyboard-interactive”. This method can be used by sysadmins to create their own authentication method. See sshd2_config(5).
- configure: If --without-ipv6 is used, checks for getipnodebyname and gethostbyname2 are disabled, i.e. those functions will not be used.
- Created new sshd2_config and ssh2_config example files as ${etcdir}/ssh{d,}2_config.example.
- Implemented XauthPath configuration option, mainly to allow the same binaries to be used both on systems that have X11 and those that do not.
- Changed the padding of packets to be done with SshRand PRNG. The protocol does not need the padding to be *that* random (it used ssh_get_random_byte(), which is overkill). There is also code to use AES as the padding generator, but you need to modify the sources (trcommon.c) for that, if you wish to play with it.
- Implemented support for /etc/nologin_<hostname>, where the filename depends on the server’s hostname. Useful with clustered machines.
- Updated sshfilexfer to SFTPv3 to conform to draft-ietf-secsh-filexfer-02.
- Implemented a SOCKS5 server to ssh2 client (you can use it as SOCKS server for other applications, and the client will create local port forwardings based on the SOCKS transaction). Create them with

  ```
  % ssh2 -L socks/<port> <remote_host>
  ```

3.2 Bug Fixes

In 3.2.9:

- Critical security fix: ASN.1 buffer overflow correction.
- RSA certificate signature fix. Two new configuration options: Cert.RSA.Compat.HashScheme and DisableVersionFallback. These settings have to be changed by editing the sshd2_config file; they cannot be altered by using the GUI.

In 3.2.5:

- Security fix: RSA signature verification security fix.
- sftp2: Fixed a bug with read line jamming when pressing backspace on AIX.

In 3.2.3:

- scp2: Removed broken special handling for SIGHUP, so that "nohup" can again work.
- ssh2: Check whether we should ignore SIGQUIT, SIGINT, and do so, if necessary. Thanks for J. Schilling for pointing this one out.
- ssh-add2: Make sure fgets() from pipe to ssh-askpass2 recovers from if interrupted by signal, i.e. SIGCHLD.
- ssh2 (lib/sshsession/sshtty.c): As entry above, but for tcsetattr().
- During "make install", use default size of key instead of hard coded 1024 when generating the host key.
- scp2,sftp2: Print progress output to stdout, to make it distinguishable from errors in cron jobs etc.
- apps/ssh/sshchsession.c: Fixed a bug that caused sshd2 child server to jam occasionally after logging an event, if nsswitch had been configured to use LDAP.
- sshd2: Fixed a bug where specifying a local forwarding endpoint as an IP-address which was irresolvable would result in a crash.
- scp2: Fixed a bug/missing feature from scp2. It now reports information also when run when there is no tty. Also implemented --statistics=[no,yes,simple], where "yes" is old-style, "no" is analogous to "-Q" command-line option, and "simple" is the way the statistics are printed when there is no tty (no intermittent reporting, file size, transfer time and full file name are printed after the transfer for the specific file is finished).
- ssh-keygen2: respect "-P" and "-p" options when converting ssh1-keys.
- lib/sshutil/sshcore/sshdebug.c: Fixed a compilation problem manifested on older AIX and debugging enabled (as is default).
- Removed ssh-pubkeymgr and ssh-chrootmgr from the distribution (they didn't work too well).
- apps/ssh/lib/sshproto/trcommon.c: Fixed a crash if hostkey algorithms or kex-methods couldn't be negotiated.
- lib/sshapputil/sshuserfile.c: Changed to use lib/sshsession/sigchld.c, instead of using wait() directly. This fixes the bug where the number of connections would slowly rise to the maximum when using MaxConnections and tcp-wrappers (it was a race condition).
- lib/sshsession/sigchld.c: Sigchld now keeps a list of recently exited children. This fixes a race condition, where the child process could exit before the mother process had registered a handler for it.
- lib/sshsession/sshunixuser.c: Make sure we have room for the NULL pointer in the group’s array.

- ssh2 (ssh1-emulation): Fixed a bug, which in some cases caused an assertion failure later.
- configure: Added /usr/X11R6/bin and /usr/X11/bin to search PATH for xauth to ease installation on pristine systems.
- lib/sshutil/sshnet/sshtcp.c: Fixed a bug with SOCKS handling.
- lib/sshutil/sshpacketstream/sshpacketwrapper.c: Fixed a latent (in ssh2) bug, when writing to the stream from the received_cb.
- lib/sshutil/sshnet/sshsocks.c: Decode ipv6-mapped-ipv4-addresses when doing SOCKS4, as SOCKS4 only supports plain ipv4-addresses.
- sshd2: Fixed a bug with originator-pat with ForwardACLs.
- scp2, sftp2: Fixed a bug, which caused file transfer to stall, if trying to transfer a zero sized file with ASCII transfer (newline mangling).
- sftp2: "ls" works much better now. Tab completion understands directories (appends a '/', for easier directory traversal).
- ssh2: Fixed a bug with one-shot forwarding.

In 3.2.2:

- Unix server VU#740619 - the setsid() issue: If executing a command without a pty (including running commands and subsystems) the child process remains in the process group of the master process, which could lead to a root compromise on platforms relying on getlogin() (mainly BSD variants). On those platforms malicious users can at least cause misleading messages to be sent to syslog and others applications.

In 3.2.0:

- ssh1-fallback mode: Fixed a problem with the stdin file descriptor being in non-blocking mode when we were executing ssh1. If ssh1 had to query something from user (e.g. for whether to store a new hostkey, etc), read() calls just returned, and ssh1 treated that as user cancelling the connection.
- Fixed a bug in the CompressionLevel configuration parameter.
- The SIGPIPE signal is handled correctly now (no warnings, error code 0, etc). We get proper disconnect messages even if the other side has closed the connection.
- ssh1 internal emulation: Fixed a bug writing new hostkeys to disk (it tried to write them to the global hostkeys directory, instead of the user's .ssh2 directory).
- Sanitized ASCII file transfers as well as minor bug fixes. This applies to all file transfer binaries (scp2, sftp2, and sftp-server).
- Fixed a bug that caused the sftp cwd to be interpreted as a regex.
- Fixed compilation for 64-bit platforms somewhat (at least on NetBSD/alpha does not complain about "unaligned access" anymore). Add a SSH_MSG_IGNORE packet to outgoing buffer, if it is empty, before appending a new packet. This is to counter the attack against predictable IVs

in CBC-mode (the attack was theoretical to begin with, so you do not need to rush your upgrades).

- Applied patch for zlib “double-free” problem. Looked it up from zlib-1.1.4. (actually, it was safe even without this patch, because we correctly check the return value of inflate(). In addition, ssh_xmalloc() and ssh_xfree() do extra checks, which also detect this kind of bugs, if you have debugging enabled.)
- Fixed some bugs in PAM authentication.

3.3 Known Issues

In 3.2.9:

- None.

In 3.2.5:

- None.

In 3.2.3:

- Tru64 cannot do chroot properly when compiled with SIA support.

In 3.2.2:

- IPv6 link-local addresses do not work
- Backspace not working in sftp window in a nested connection.

In 3.2.1:

- SFTP-logging does not work with chrooted accounts on AIX.

In 3.2.0:

- ‘make clean’ and ‘make distclean’ result in an error on BSD platforms.
- Password aging is not functional on Solaris 8.
- Issues with IPv6 addresses and ‘RequireReverseMapping’ using IPv6.

4 Windows Client

The Windows binaries are included only in the *SSH Secure Shell for Workstations* product.

4.1 New Features

In 3.2.9.2:

- None.

In 3.2.9:

- None.

In 3.2.5:

- None.

In 3.2.3:

GUI client

- Possibility to select 'No to All' in overwrite confirmation dialog.
- Possibility to disconnect when file transfer is active.

Command Line Clients

- Architecture change for scp2 and sftp2. Direct port from Unix. scp2 and sftp2 launch ssh2 underneath. scp2 and sftp2 has the same full functionality as on Unix.
- Architecture change brings all ssh2 connection features to scp2 and sftp2.
- ssh2: verbose debug output was improved & cleaned.
- scp2: option -m sets both default file and dir permission bits for upload. (Does not talk about permission mask, which it is not.)
- sftp2: command setperm sets both default file and dir permission bits for upload.
- ssh2 & scp2 & sftp2: new option '-k' for specifying a custom configuration data dir, required in scheduler jobs.
- ssh2: if the identification file does not exist, all the user keys under userkeys dir are taken as candidates to the login.

In 3.2.2:

- None

In 3.2.1:

- None

In 3.2.0:

File transfer:

- Local view: displays local drives and enables easy drag and drop between local and remote computers.
- Three file transfer window layouts.
- Transfer view: displays transferred files and keeps a list of custom transfer queues.
- File bar: displays current folder and keeps a list of user's favourite folders.
- Terminal and file transfer windows can be opened in the current working directory.
- Multiple simultaneous transfers.

Other new features:

- Integration with Accession, an authentication agent.
- Generic authentication method called Keyboard-Interactive.
- Position of windows can be saved and windows can be restored in their configured position on demand.
- Host and user name can be requested from user when a profile is selected.
- Improved configuration pages.
- Configuration helper for command line client, ssh2.exe
- Creates a desktop shortcut to a profile.
- Partial SOCKS5 support.

4.2 Bug Fixes

In 3.2.9.2:

- Silent installation did not work.
- X11 tunneling did not work.
- SSH1 usage caused the client to crash.

In 3.2.9:

- Critical security fix: ASN.1 buffer overflow correction.
- RSA certificate signature fix. Now it is possible to define the hash format used and to disable fallback compatibility code by utilizing new configuration file options, Cert.RSA.Compat.HashScheme (with valid values of md5 and sha1) and DisableVersionFallback, respectively. These settings have to be changed by editing the global.dat configuration file; they cannot be altered by using the GUI.

In 3.2.5:

- Security fix: RSA signature verification security fix.

In 3.2.3:

GUI client

- If client has CA certificates, it prefers server certificates to pubkeys for server authentication.
- When 'preserve original file time' is not checked, local file edit doesn't ask to upload the file when it hasn't been modified anymore.
- Organizational unit and Common name have correct order in certificate enrolment
- Closing terminal window doesn't crash client after idle timeout from server.
- Client doesn't allow two or more file transfers to update the same slot in the file transfer list.
- Client doesn't crash when file transfer window is closed and active file transfers exist.
- Client doesn't crash when disconnected twice.
- Terminal using the ANSI with csh doesn't crash when backspacing long text.
- Default user path is added to the debug file name if user doesn't specify path.
- Debug files over 3 MB won't be opened in debug view.
- Opening folders that do not exist does not cause the client to crash anymore.
- Localview remembers column used in sorting when directory changes.
- Cancelling file transfer when terminal window is flooded with text doesn't cause client to crash anymore.
- Profiles are sorted alphabetically on every platform.

- If error occurs during file transfer other files in the same transfer will be marked as 'error' not 'queued'.
- Client doesn't crash anymore when a lot of text without newlines is output to the screen and mouse is moved.
- Focus doesn't stay on menu button when pressed twice in a row.
- Client doesn't crash anymore on XP when file transfer frame is opened after a certain amount of time.
- Time left field doesn't show negative numbers before file transfer starts.
- Red underscores do not appear anymore on terminal screen.
- Client doesn't crash anymore when trying to transfer file with ':' in their names.
- Editing locally multiple files with same names doesn't cause uploading the files to wrong places anymore.
- Arrows in file transfer bar and in remote viewpoint to correct direction when sorting lists by size.
- In file transfer bar KB/s -> kB/s.
- User is prompted if active file transfers should be cancelled when file transfer window is closed.

Command Line Clients

- ssh2: Attempt to sign using key on PKCS #11 token caused crash.
- ssh2: Crashed when PKCS #11 enabled and key exchange fails.
- ssh2: The default configuration data was wrong on Win9x. Now it is under <INSTALLDIR>\..\Users\%U, like it is with the GUI client. Hostkeys and userkeys can be thus shared.
- scp2: declines to support undocumented options '4' '6' 'W' 'n'.
- sftp2: does not show unsupported options '4' '6' in usage.
- ssh2: Piping through ssh2 works. ("dir | ssh2 host wc")
- ssh2: default cipher is aes and not 3des.
- ssh2: does not go fatal if a timeout occurs in Accession or EK while waiting for user PIN and if user then enters correct soft passhprase and PIN.
- ssh2: does not crash if active tunnels exist and a key is pressed on Win9x.
- ssh2: declines the use of Unix-specific options 'f' 'n' 'P' '8' 'k' '1' '6' '4'.
- ssh2: software certificates work.
- ssh2: slashes are not interpreted as an option prefix. (ssh2 host "/bin/sh" works)
- ssh2: SSH1 works. (Earlier client went to fatal.)
- ssh2: If option -S was used, it was not possible to use CTRL+C to exit client.

- sftp2&scp2: exit codes now work and are identical to Unix.
- sftp2&scp2: files with "()[]#" in name can now be copied.
- sftp2&scp2: the regex patterns work and are case insensitive, both local and remote end. Only * and ? valid patterns in Windows.
- sftp2&scp2: more efficient copying. Does not read unnecessary directories. (This only appeared as slowness, no big bug.)
- sftp2&scp2: password is no longer accepted from a file for security reasons.
- sftp2: went to fatal while normal download.
- sftp2: lcd works to any local drive and network share.
- sftp2: pager correctly sees the height of the console window. (Previously just scrolled only 10 lines at a time.)
- ssh2: Fixed a crash if hostkey algorithms or kex-methods couldn't be negotiated.
- scp2 & sftp2: Print progress output to stdout, to make it distinguishable from errors in cron jobs etc.

In 3.2.2:

- Security fix: URL link security fix. Fixed a buffer overflow in URL handling code. When the user was provided a long URL in the terminal window, and the user clicked the URL, it overflowed in memory during the resulting copy operation. All language Versions 3.1.0, 3.1.1 and 3.2.0 of the Windows client were affected.

In 3.2.1:

- AIX patch release: chrooting was not functioning properly in the AIX binaries and source distribution of 3.2.0. Other platforms are not affected.

In 3.2.0:

- Security fix: CBC-IV attack has been eliminated.
- FTP tunnelling.
- SFTP ASCII transfer.
- Banner message is always displayed before authentication.
- If only tunnels are requested, does not disconnect when tunnel closes.
- Command line client ssh2.exe used wrong user settings directory on Win9x.

4.3 Known Issues

In 3.2.9.2 - 3.2.3:

- GUI: Using (slightly) incorrect certificate in the client side for the Server Authentication hangs the client if the server identification has both the public key and the certificate authentication enabled. This does not happen, however, when using an invalid certificate or there is no server certificate when connecting to the server.
- SSH2.EXE with an external key provider: Crash might occur if PKCS #11 is configured in the ssh2_config and key exchange fails, for example because user does not wish to save the hostkey.
- SSH2.EXE with an external key provider (eToken): Using PKCS #11 via SSH Accession results a server to reject a signature.
- SFTP2.EXE, SCP2.EXE: Full new/changed host key dialog not displayed to the user.

In 3.2.2:

- Crashing of client when resuming from hibernation (with laptops mostly).
- SFTP GUI: transfer list status shows %-bar even after connection lost.
- SFTP GUI: local view: network share closed after downloading.
- WIN9X: SSH2 command line: crashing when exiting client (FTP tunnelled).
- Sorting order is different in folder and file lists.
- Importing (not valid) large dat file as licence will crash client.
- Using the download dialog to download a file to a non-mapped network share fails in the SFTP client.

In 3.2.0:

- Modifying ‘keymap22.map’ incorrectly crashes the Windows client.
- Windows command line tool ‘ssh2’ uses 3DES as the default cipher instead of AES128.
- Windows command line tool ‘ssh2’ fails to authenticate a user using certificates.
- Disabling a PKCS #11 provider in the Windows client does not function correctly.
- When using patterns such as “?” and “*” to match a set of filenames in the Windows version of ‘scp2’, scp2 fails to work as expected.
- Virtual drives are not displayed in the Windows SFTP client.
- A ‘:’ character in a filename causes SFTP Client to crash.
- Editing files with the same filename via the SFTP client for Windows may overwrite the wrong file.

- If a directory’s name starts with ‘#’, an attempted SFTP listing of it produces an ‘invalid pattern’ error.
- The User Profile Location is determined incorrectly in the Windows client.
- A ‘Corrupted MAC on input’ protocol error is occasionally produced when uploading using SFTP.
- URLs are incorrectly parsed in the Windows client.
- 1-byte file upload using text mode causes a crash on certain platforms.
- It is possible to ‘lcd’ in the Windows command line tool ‘sftp2’ only once if the destination directory is located on a different drive.
- Windows command line tools ‘sftp2’ and ‘scp2’ return errorlevel 0 even upon disconnection, or upon child process ‘ssh2’ returning another value.