*Open*SSH Release Notes

OpenSSH 10.3/10.3p1 (2026-04-02)

OpenSSH 10.3 was released on 2026-04-02. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
https://www.openssh.com/donations.html

Potentially-incompatible changes
----

 * ssh(1), sshd(8): remove bug compatibility for implementations
   that don't support rekeying. If such an implementation tries to
   interoperate with OpenSSH, it will now eventually fail when the
   transport needs rekeying.

 * sshd(8): prior to this release, a certificate that had an empty
   principals section would be treated as matching any principal
   (i.e. as a wildcard) when used via authorized_keys principals=""
   option. This was intentional, but created a surprising and
   potentially risky situation if a CA accidentally issued a
   certificate with an empty principals section: instead of being
   useless as one might expect, it could be used to authenticate as
   any user who trusted the CA via authorized_keys. [Note that this
   condition did not apply to CAs trusted via the sshd_config(5)
   TrustedUserCAKeys option.]

   This release treats an empty principals section as never matching
   any principal, and also fixes interpretation of wildcard
   characters in certificate principals. Now they are consistently
   implemented for host certificates and not supported for user
   certificates.

 * ssh(1): the -J and equivalent -oProxyJump="..." options now
   validate user and host names for ProxyJump/-J options passed
   via the command-line (no such validation is performed for this
   option in configuration files). This prevents shell injection in
   situations where these were directly exposed to adversarial
   input, which would have been a terrible idea to begin with.
   Reported by rabbit.

Changes since OpenSSH 10.2
==========================

This release contains some relatively minor security fixes as well
as a number of feature improvements and general bugfixes.

Security
========

 * ssh(1): validation of shell metacharacters in user names supplied
   on the command-line was performed too late to prevent some
   situations where they could be expanded from %-tokens in
   ssh_config. For certain configurations, such as those that use a
   "%u" token in a "Match exec" block, an attacker who can control
   the user name passed to ssh(1) could potentially execute arbitrary
   shell commands. Reported by Florian Kohnhäuser.

   We continue to recommend against directly exposing ssh(1) and
   other tools' command-lines to untrusted input. Mitigations such
   as this can not be absolute given the variety of shells and user
   configurations in use.

 * sshd(8): when matching an authorized_keys principals="" option
   against a list of principals in a certificate, an incorrect
   algorithm was used that could allow inappropriate matching in
   cases where a principal name in the certificate contains a
   comma character. Exploitation of the condition requires an
   authorized_keys principals="" option that lists more than one
   principal *and* a CA that will issue a certificate that encodes
   more than one of these principal names separated by a comma
   (typical CAs strongly constrain which principal names they will
   place in a certificate). This condition only applies to user-
   trusted CA keys in authorized_keys, the main certificate
   authentication path (TrustedUserCAKeys/AuthorizedPrincipalsFile)
   is not affected. Reported by Vladimir Tokarev.

 * scp(1): when downloading files as root in legacy (-O) mode and
   without the -p (preserve modes) flag set, scp did not clear
   setuid/setgid bits from downloaded files as one might typically
   expect. This bug dates back to the original Berkeley rcp program.
   Reported by Christos Papakonstantinou of Cantina and Spearbit.

 * sshd(8): fix incomplete application of PubkeyAcceptedAlgorithms
   and HostbasedAcceptedAlgorithms with regard to ECDSA keys.
   Previously if one of these directives contains any ECDSA algorithm

name (say "ecdsa-sha2-nistp384"), then any other ECDSA algorithm would be accepted in its place regardless of whether it was listed or not. Reported by Christos Papakonstantinou of Cantina and Spearbit.

* ssh(1): connection multiplexing confirmation (requested using "ControlMaster ask/autoask") was not being tested for proxy mode multiplexing sessions (i.e. "ssh -O proxy ..."). Reported by Michalis Vasileiadis.

New features
----

* ssh(1), sshd(8): support IANA-assigned codepoints for SSH agent forwarding, as per draft-ietf-sshm-ssh-agent. Support for the new names is advertised via the EXT_INFO message. If a server offers support for the new names, then they are used preferentially.

  Support for the pre-standardisation "@openssh.com" extensions for agent forwarding remains supported.

* ssh-agent(1): implement support for draft-ietf-sshm-ssh-agent "query" extension.

* ssh-add(1): support querying the protocol extensions via the agent "query" extension with a new -Q flag.

* ssh(1): support multiple files in a ssh_config RevokedHostKeys directive. bz3918

* sshd(8): support multiple files in a sshd_config RevokedKeys directive bz3918

* ssh(1): add a ~I escape option that shows information about the current SSH connection.

* ssh(1): add an "ssh -Oconninfo user@host" multiplexing command that shows connection information, similar to the ~I escapechar.

* ssh(1): add an "ssh -O channels user@host" multiplexing command to get a running mux process to show information about what channels are currently open.

* sshd(8): add 'invaliduser' penalty to PerSourcePenalties, which is applied to login attempts for usernames that do not match real accounts. Defaults to 5s to match 'authfail' but allows administrators to block such attempts for longer if desired.

* sshd(8): add a GSSAPIDelegateCredentials option for the server, controlling whether it accepts delegated credentials offered by the client. This option mirrors the same option in ssh_config. GHPR614

* ssh(1), sshd(8): support the VA DSCP codepoint in the IPQoS directive.

* sshd(8): convert PerSourcePenalties to using floating point time, allowing penalties to be less than a second. This is useful if you need to penalise things you expect to occur at >=1 QPS.

* ssh-keygen(1): support writing ED25519 keys in PKCS8 format. GHPR570

* Support the ed25519 signature scheme via libcrypto.

Bugfixes
----

* sshd(8): make IPQoS first-match-wins in sshd_config, like other configuration directives. bz3924

* sshd(8): fix potential crash when MaxStartups is using a single argument (i.e. not using the MaxStartps x:y:z form) to a value below 10. bz3941

* sshd(8): fix a potential hang during key exchange if needed DH group values were missing from /etc/moduli.

* ssh-agent(1): fix return values from extensions to be correct wrt draft-ietf-sshm-ssh-agent: extension requests should indicate failure using SSH_AGENT_EXTENSION_FAILURE rather than the generic SSH_AGENT_FAILURE error code. This allows the client to discern between "the request failed" and "the agent doesn't support this extension".

* ssh(1): use fmprintf for showing challenge-response name and info to preserve UTF-8 characters where appropriate. Prompted by GitHub PR#452.

* scp(1): when uploading a directory using sftp/sftp (e.g. during a recursive transfer), don't clobber the remote directory permissions unless either we created the directory during the transfer or the -p flag was set. bz3925

* All: implement missing pieces of FIDO/webauthn signature support, mostly related to certificate handling and enable acceptance of this signature format by default. bz3748 GHPR624 GHPR625

* sshd_config(5): make it clear that DenyUsers/DenyGroups overrides AllowUsers/AllowGroups. Previously we specified the order in which the directives are processed but it was ambiguous as to what happened if both matched.

* ssh(1): don't try to match certificates held in an agent to private keys. This matching is done to support certificates that were loaded without their private key material, but is unnecessary for agent-hosted certificate which always have private key material available in the agent. Worse, this matching would mess up the request sent to the agent in such a way as to break usage of these keys when the key usage was restricted in the agent. bz3752

* sftp(1): if editline has been switched to vi mode (i.e. via "bind -v" in .editrc), setup a keybinding so that command mode can be entered.

* ssh(1), sshd(8): improve performance of keying the sntrup761 key agreement algorithm.

* ssh(1), sshd(8): enforce maximum packet/block limit during pre-authentication phase.

* sftp(1): don't misuse the sftp limits extension's open-handles field. This value is supposed to be the number of handles a server will allow to be opened and not a number of outstanding read/write requests that can be sent during an upload/download.

* sshd(8): don't crash at connection time if the main sshd_config lacks any subsystem directive but one is defined in a Match block. bz3906

* sshd_config(5): add a warning next to the ForceCommand directive that forcing a command doesn't automatically disable forwarding.

* sshd_config(5): add a warning that TOKENS are replaced without filtering or escaping and that it's the administrator's responsibility to ensure they are used safely in context.

* scp(1): correctly quote filenames in verbose output for local-> local copies. bz3900

* sshd(8): don't mess up the PerSourceNetBlockSize IPv6 mask if sscanf didn't decode it. GHPR598

* ssh-add(1): when loading FIDO2 resident keys, set the comment to the FIDO application string. This matches the behaviour of ssh-keygen -K. GHPR608

* sshd(8): don't strnvis() log messages that are going to be logged by sshd-auth via its parent sshd-session process, as the parent will also run them though strnvis(). Prevents double-escaping of non-printing characters in some log messages. bz3896

* ssh-agent(1): escape SSH_AUTH_SOCK paths that are sent to the shell as setenv commands. Unbreaks ssh-agent for home directory paths that contain whitespace. bz3884

* All: Remove unnecessary checks for ECDSA public key validity.

* sshd(8): activate UnusedConnectionTimeout only after the last channel has closed. Previously UnusedConnectionTimeout could fire early after a ChannelTimeout. This was not a problem for the OpenSSH client because it terminates once all channels have closed but could cause problems for other clients (e.g. API clients) that do things differently. bz3827

* All: fix PKCS#11 key PIN entry problems introduced in openssh-10.1/10.2. bz3879

* scp(1): when using the SFTP protocol for transfers, fix implicit destination path selection when source path ends with "..". bz3871

* sftp(1): when tab-completing a filename, ensure that the completed string does not end up mid-way through a multibyte character, as this will cause a fatal() later on. GHPR#587

* ssh-keygen(1): fix crash at exit (visible via ssh-keygen -D) when multiple keys loaded.

* scp(1)/sftp(1): correctly display bandwidths >2GBps in the progress meter.

Portability
---

* sshd(8): fix condition introduced in openssh 10.2p1 stable branch here a PAM module that changed the requested username between SSH_MSG_USERAUTH_REQUEST messages during authentication could confuse the PAM stack and let it proceed with a different understanding of the active username than the rest of sshd. Reported by Mike Damm.

* sshd(8): immediately report interactive instructions to clients when using keyboard-interactive authentication with PAM. bz2876

* sshd(8): fix duplicate PAM messages under some situations.

```
 * sshd(8): don't leak PAM handle on repeat invocations. bz3882

 * All: support linking libcrypto implementations (e.g. BoringSSL)
   that require libstdc++.

 * sshd(8): fix ut_type for btmp records, correctly using
   LOGIN_PROCESS and USER_PROCESS.

 * sshd(8): allow uname(3) in the seccomp sandbox. This is needed by
   zlib-ng on RISC-V platforms.

 * All: remove remaining OpenSSL_add_all_algorithms() calls.
   We already have OPENSSL_init_crypto() in the compat layer.
   Prompted by github PR#606

 * All: fix builds on older Mac OS wrt nfds_t.

 * mdoc2man: several improvements including better support for Dl
   and Ns inside Ic.

Checksums:
==========

 - SHA1 (openssh-10.3.tar.gz) = 854863c04cd28242d73ac6c3ee9c37fa756f1a2f
 - SHA256 (openssh-10.3.tar.gz) = aCU5P47rM+m4N8/i2JOHMOMhafMYqBhvQQSnPXczN5M=

 - SHA1 (openssh-10.3p1.tar.gz) = 9c78838ec07af14aff54f3755ac56ce6812452a9
 - SHA256 (openssh-10.3p1.tar.gz) = VmgqNruS3PS08Bb9jsjnQFm3mo3iXBXWcNcx59GORfQ=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available from the mirror sites:
https://cdn.openbsd.org/pub/OpenBSD/OpenSSH/RELEASE_KEY.asc

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com
```

OpenSSH 10.2/10.2p1 (2025-10-10)

```
OpenSSH 10.2 was released on 2025-10-10. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
https://www.openssh.com/donations.html

Future deprecation warning
--------------------------

 * A future release of OpenSSH will deprecate support for SHA1 SSHFP
   records due to weaknesses in the SHA1 hash function. SHA1 SSHFP
   DNS records will be ignored and ssh-keygen -r will generate only
   SHA256 SSHFP records.

   The SHA256 hash algorithm, which has no known weaknesses, has
   been supported for SSHFP records since OpenSSH 6.1, released in
   2012.

Changes since OpenSSH 10.1
==========================

This is a bugfix release, primarily to fix a problem that rendered
ssh(1) unusable when ControlPersist was enabled.

Bugfixes
--------

 * ssh(1): fix mishandling of terminal connections when
   ControlPersist was active that rendered the session unusable.
   bz3872

 * ssh-keygen(1): fix download of keys from PKCS#11 tokens.

 * ssh-keygen(1): fix CA signing operations when the CA key is held
   in a ssh-agent(1). bz3877


Portability
-----------

 * All: support platforms without mmap(2), e.g. WASM builds such as
   https://hterm.org

 * All: fix builds on FreeBSD for missing fnctl.h include.

 * All: fix builds on MacOS <10.12 Sierra, which lacks
   clock_gettime(3)
```

* sshd(8): don't PAM_RHOST if the remote host is the "UNKNOWN"
  placeholder name. Avoids potential hangs in some PAM modules as
  they try to resolve it. Note, sshd(8) only uses the "UNKNOWN"
  name when the connection is not on an IPv4 or IPv6 socket.

Checksums:
==========

SHA1 (openssh-10.2.tar.gz) = 6fcda8004bad0fb0eaee60e8308f91b605ad0dce
SHA256 (openssh-10.2.tar.gz) = y0rCEdrVc4OJRZLg0u3F0frAgz87ydeTktCk3rQfVj8=

SHA1 (openssh-10.2p1.tar.gz) = c34efade16109f065ec8c834f237bcedd8d7ef5c
SHA256 (openssh-10.2p1.tar.gz) = zMQsBBmTeVkmP6Hb0W2vwYxWuYTANWLSk3zlamD3mLI=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available from the mirror sites:
https://cdn.openbsd.org/pub/OpenBSD/OpenSSH/RELEASE_KEY.asc

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

---

OpenSSH 10.1/10.1p1 (2025-10-06)

OpenSSH 10.1 was released on 2025-10-06. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
https://www.openssh.com/donations.html

Future deprecation warning
----

 * A future release of OpenSSH will deprecate support for SHA1 SSHFP
   records due to weaknesses in the SHA1 hash function. SHA1 SSHFP
   DNS records will be ignored and ssh-keygen -r will generate only
   SHA256 SSHFP records.

   The SHA256 hash algorithm, which has no known weaknesses, has
   been supported for SSHFP records since OpenSSH 6.1, released in
   2012.

Potentially-incompatible changes
----

 * ssh(1): add a warning when the connection negotiates a non-post
   quantum key agreement algorithm.

   This warning has been added due to the risk of "store now, decrypt
   later" attacks. More details at https://openssh.com/pq.html

   This warning may be controlled via a new WarnWeakCrypto ssh_config
   option, defaulting to on. This option is likely to control
   additional weak crypto warnings in the future.

 * ssh(1), sshd(8): major changes to handling of DSCP marking/IPQoS

   In both client and server the default DSCP (a.k.a IPQoS) values
   were revised and the way these values are used during runtime has
   changed.

   Interactive traffic is now assigned to the EF (Expedited
   Forwarding) class by default. This provides more appropriate
   packet prioritisation information for the intermediate network,
   such as wireless media (cf. RFC 8325). Non-interactive traffic
   will now use the operating system default DSCP marking. Both the
   interactive and non-interactive DSCP values may be overridden via
   the IPQoS keyword, described in ssh_config(5) and sshd_config(5).

   The appropriate DSCP marking is now automatically selected and
   updated as needed over the course of a connection's lifetime.
   ssh(1) and sshd(8) will switch between the interactive and
   non-interactive IPQoS values depending on the type of SSH
   channels open at the time. For example, if an sftp session is
   using the connection alongside a shell session, then the non-
   interactive value will be used for the duration of the sftp. A
   connection which contains only interactive sessions is marked EF.

 * ssh(1), sshd(8): deprecate support for IPv4 type-of-service (ToS)
   keywords in the IPQoS configuration directive.

   Type of Service (ToS) was deprecated in the late nineties and
   replaced with the Differentiated Services architecture, which
   has significant advantages for operators because it offers more
   granularity.

   OpenSSH switched its default IPQoS from ToS to DSCP values in

```
    2018 (openssh-7.7).

    IPQoS configurations with 'lowdelay', 'reliability', or
    'throughput' will be ignored and will instead use the system
    default QoS settings. Additionally, a debug message will be logged
    about the deprecation with a suggestion to use DSCP QoS instead.

  * ssh-add(1): when adding certificates to an agent, set the expiry
    to the certificate expiry time plus a short (5 min) grace period.

    This will cause the agent to automatically remove certificates
    shortly after they expire. A new ssh-add -N option disables this
    behaviour.

  * All: remove experimental support for XMSS keys. This was never
    enabled by default. We expect to implement a new post-quantum
    signature scheme in the near future.

  * ssh-agent(1), sshd(8): move agent listener sockets from /tmp to
    under ~/.ssh/agent for both ssh-agent(1) and forwarded sockets
    in sshd(8).

    This ensures processes that have restricted filesystem access
    that includes /tmp do not ambiently have the ability to use keys
    in an agent.

    Moving the default directory has the consequence that the OS will
    no longer clean up stale agent sockets, so ssh-agent now gains
    this ability.

    To support $HOME on NFS, the socket path includes a truncated
    hash of the hostname. ssh-agent will, by default, only clean up
    sockets from the same hostname.

    ssh-agent(1) gains some new flags: -U suppresses the automatic
    cleanup of stale sockets when it starts. -u forces a cleanup
    without keeping a running agent, -uu forces a cleanup that ignores
    the hostname. -T makes ssh-agent put the socket back in /tmp.

Changes since OpenSSH 10.0
==========================

This release contains a minor security fix as well as a number of
feature improvements and bugfixes.

Security
========

* ssh(1): disallow control characters in usernames passed via the
  commandline or expanded using %-sequences from the configuration
  file, and disallow \0 characters in ssh:// URIs.

  If an ssh(1) commandline was constructed using usernames or URIs
  obtained from an untrusted source, and if a ProxyCommand that uses
  the %r expansion was configured, then it may be possible for an
  attacker to inject shell expressions that may be executed when the
  proxy command is started.

  We strongly recommend against using untrusted inputs to construct
  ssh(1) commandlines.

  This change also relaxes the validity checks in one small way:
  usernames supplied via the configuration file as literals (i.e.
  that have no % expansion characters) are not subject to these
  validity checks. This allows usernames that contain arbitrary
  characters to be used, but only via configuration files. This is
  done on the basis that ssh's configuration is trusted.

  This issue was reported by David Leadbeater.

New features
------------

  * ssh(1), sshd(8): add SIGINFO handlers to log active channel and
    session information.

  * sshd(8): when refusing a certificate for user authentication, log
    enough information to identify the certificate in addition to the
    reason why it was being denied. Makes debugging certificate
    authorisation problems a bit easier.

  * ssh(1), ssh-agent(1): support ed25519 keys hosted on PKCS#11
    tokens.

  * ssh(1): add an ssh_config(5) RefuseConnection option that, when
    encountered while processing an active section in a
    configuration, terminates ssh(1) with an error message that
    contains the argument to the option.

    This may be useful for expressing reminders or warnings in config
    files, for example:

    Match host foo
             RefuseConnection "foo is deprecated, use splork instead"

  * sshd(8): make the X11 display number check relative to
    X11DisplayOffset. This will allow people to use X11DisplayOffset
    to configure much higher port ranges if they really want, while
```

not changing the default behaviour.

* unit tests: the unit test framework now includes some basic benchmarking capabilities. Run with "make UNITTEST_BENCHMARK=yes" on OpenBSD or "make unit-bench" on Portable OpenSSH.

Bugfixes
----

* sshd(8): fix mistracking of MaxStartups process exits in some situations. At worst, this could cause all MaxStartups slots to fill and sshd to refuse new connections.

* ssh(1): fix delay on X client startup when ObscureKeystrokeTiming is enabled. bz#3820

* sshd(8): increase the maximum size of the supported configuration from 256KB to 4MB, which ought to be enough for anybody. Fail early and visibly when this limit is breached. bz3808

* sftp(1): during sftp uploads, avoid a condition where a failed write could be ignored if a subsequent write succeeded. This is unlikely but technically possible because sftp servers are allowed to reorder requests.

* sshd(8): avoid a race condition when the sshd-auth process exits that could cause a spurious error message to be logged.

* sshd(8): log at level INFO when PerSourcePenalties actually blocks access to a source address range. Previously this was logged at level VERBOSE, which hid enforcement actions under default config settings.

* sshd(8): GssStrictAcceptor was missing from sshd -T output; fix

* sshd(8): Make the MaxStartups and PerSourceNetBlockSize options first-match-wins as advertised. bz3859

* ssh(1): fix an incorrect return value check in the local forward cancellation path that would cause failed cancellations not to be logged.

* sshd(8): make "Match !final" not trigger a second parsing pass of ssh_config (unless hostname canonicalisation or a separate "Match final" does). bz3843

* ssh(1): better debug diagnostics when loading keys. Will now list key fingerprint and algorithm (not just algorithm number) as well as making it explicit which keys didn't load.

* All: fix a number of memory leaks found by LeakSanitizer, Coverity and manual inspection.

* sshd(8): Output the current name for PermitRootLogin's "prohibit-password" in sshd -T instead of its deprecated alias "without-password". bz#3788

* ssh(1): make writing known_hosts lines more atomic by writing the entire line in one operation and using unbuffered stdio.

  Usually writes to this file are serialised on the "Are you sure you want to continue connecting?" prompt, but if host key checking is disabled and connections were being made with high concurrency then interleaved writes might have been possible.

Portability
----

* sshd(8): check the username didn't change during the PAM transactions.

  PAM modules can change the user during their execution, but this is not supported by sshd(8). If such a case was incorrectly configured by the system administrator, then sshd(8) could end up using a different username to the one authorised by PAM.

* sshd(8): don't log audit messages with UNKNOWN hostname to avoid slow DNS lookups in the audit subsystem.

* All: when making a copy of struct passwd, ensure struct fields are non-NULL. Android libc can return NULL pw_gecos, for example.

* All: Remove status bits from OpenSSL >=3 version check.

* sshd(8), ssh(1): Use SSH_TUN_COMPAT_AF on FreeBSD. Otherwise tun forwarding from other OSes fails as soon as the first IPv6 message is sent by the other side (which is usually a Router Solicitation ICMPv6 message which is sent as soon as the interface is up).

* ssh(1), ssh-agent(8): check for nlist function presence before attempting to use it instead of relying on the presence of the nlist.h header. Mac OS X, for example, has the header but not the function in the 64bit libraries.

* All: fill in missing system header files.

  Create replacement header files inside openbsd-compat for common headers that are missing on a given platform. Usually these are

```
    just empty, but in some cases they'll include the equivalent file.
    This avoids having to wrap those includes in '#ifdef HAVE_FOO_H'
    and reduces the diff between Portable OpenSSH and OpenBSD.

  * sshd(8): handle futex_time64 properly in seccomp sandbox
    Previously we only allowed __NR_futex, but some 32-bit systems
    apparently support __NR_futex_time64. We had support for this
    in the sandbox, but because of a macro error only __NR_futex was
    allowlisted.

  * Add contrib/gnome-ssh-askpass4 for GNOME 40+ using the GCR API.

  * sshd(8): let ga_init() fail gracefully if getgrouplist does.
    Apparently getgrouplist() can fail on OSX when passed a
    non-existent group name. Other platforms seem to return a group
    list consisting of the numeric gid passed to the function. bz3848

  * ssh-agent(1): exit 0 from SIGTERM under systemd socket-activation,
    preventing a graceful shutdown of an agent via systemd from
    incorrectly marking the service as "failed".

  * build: wrap some autoconf macros in AC_CACHE_CHECK.

    This allows skipping/overriding the OSSH_CHECK_CFLAG_COMPILE and
    OSSH_CHECK_CFLAG_LINK macros used to discover supported compiler
    or linker flags. E.g.

      $ ./configure ossh_cv_cflag__fzero_call_used_regs_used=no
      [...]
      checking if cc supports compile flag -fzero-call-used-regs=used
      and linking succeeds... (cached) no

Checksums:
==========

SHA1 (openssh-10.1.tar.gz) = 8eef44a945a9a9a5a99213ab0d57e35b7ba60e75
SHA256 (openssh-10.1.tar.gz) = j9ymvhvdGMeAvh1oTI2YmOAwv7Ao70gbGPyPedgQsBU=

SHA1 (openssh-10.1p1.tar.gz) = 7fd17b99d1beffb47cd380d64079e920bb0bd91f
SHA256 (openssh-10.1p1.tar.gz) = ufx6K4JXlGem8vQ+SoHI4d/aYU3bT5slWq/XAgu/B1g=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available from the mirror sites:
https://cdn.openbsd.org/pub/OpenBSD/OpenSSH/RELEASE_KEY.asc

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com
```

OpenSSH 10.0/10.0p2 (2025-04-09)

```
OpenSSH 10.0 was released on 2025-04-09. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
https://www.openssh.com/donations.html

Potentially-incompatible changes
--------------------------------

 * This release removes support for the weak DSA signature
   algorithm, completing the deprecation process that began in
   2015 (when DSA was disabled by default) and repeatedly warned
   over the last 12 months.

 * scp(1), sftp(1): pass "ControlMaster no" to ssh when invoked by
   scp & sftp. This disables implicit session creation by these
   tools when ControlMaster was set to yes/auto by configuration,
   which some users found surprising. This change will not prevent
   scp/sftp from using an existing multiplexing session if one had
   already been created. GHPR557

 * This release has the version number 10.0 and announces itself
   as "SSH-2.0-OpenSSH_10.0". Software that naively matches
   versions using patterns like "OpenSSH_1*" may be confused by
   this.

 * sshd(8): this release removes the code responsible for the
   user authentication phase of the protocol from the per-
   connection sshd-session binary to a new sshd-auth binary.
   Splitting this code into a separate binary ensures that the
   crucial pre-authentication attack surface has an entirely
   disjoint address space from the code used for the rest of the
   connection. It also yields a small runtime memory saving as the
   authentication code will be unloaded after the authentication
   phase completes. This change should be largely invisible to
   users, though some log messages may now come from "sshd-auth"
```

instead of "sshd-session". Downstream distributors of OpenSSH will need to package the sshd-auth binary.

* sshd(8): this release disables finite field (a.k.a modp) Diffie-Hellman key exchange in sshd by default. Specifically, this removes the "diffie-hellman-group*" and "diffie-hellman-group-exchange-*" methods from the default KEXAlgorithms list. The client is unchanged and continues to support these methods by default. Finite field Diffie Hellman is slow and computationally expensive for the same security level as Elliptic Curve DH or PQ key agreement while offering no redeeming advantages. ECDH has been specified for the SSH protocol for 15 years and some form of ECDH has been the default key exchange in OpenSSH for the last 14 years.

* sshd(8): this release removes the implicit fallback to compiled-in groups for Diffie-Hellman Group Exchange KEX when the moduli file exists but does not contain moduli within the client-requested range. The fallback behaviour remains for the case where the moduli file does not exist at all. This allows administrators more explicit control over which DH groups will be selected, but can lead to connection failures if the moduli file is edited incorrectly. bz#2793

Changes since OpenSSH 9.9
=========================

This release contains a minor security fix as well as a number of feature improvements and bugfixes.

Security
========

* sshd(8): fix the DisableForwarding directive, which was failing to disable X11 forwarding and agent forwarding as documented. X11 forwarding is disabled by default in the server and agent forwarding is off by default in the client.

New features
----

* ssh(1): the hybrid post-quantum algorithm mlkem768x25519-sha256 is now used by default for key agreement. This algorithm is considered to be safe against attack by quantum computers, is guaranteed to be no less strong than the popular curve25519-sha256 algorithm, has been standardised by NIST and is considerably faster than the previous default.

* ssh(1): prefer AES-GCM to AES-CTR mode when selecting a cipher for the connection. The default cipher preference list is now Chacha20/Poly1305, AES-GCM (128/256) followed by AES-CTR (128/192/256).

* ssh(1): add %-token and environment variable expansion to the ssh_config SetEnv directive.

* ssh(1): allow %-token and environment variable expansion in the ssh_config User directive, with the exception of %r and %C which would be self-referential. bz#3477

* ssh(1), sshd(8): add "Match version" support to ssh_config and sshd_config. Allows matching on the local version of OpenSSH, e.g. "Match version OpenSSH_10.*".

* ssh(1): add support for "Match sessiontype" to ssh_config. Allows matching on the type of session initially requested, either "shell" for interactive sessions, "exec" for command execution sessions, "subsystem" for subsystem requests, such as sftp, or "none" for transport/forwarding-only sessions.

* ssh(1): add support for "Match command ..." support to ssh_config, allowing matching on the remote command as specified on the command-line.

* ssh(1): allow 'Match tagged ""' and 'Match command ""' to match empty tag and command values respectively.

* sshd(8): allow glob(3) patterns to be used in sshd_config AuthorizedKeysFile and AuthorizedPrincipalsFile directives. bz2755

* sshd(1): support the VersionAddendum in the client, mirroring the option of the same name in the server; bz2745

* ssh-agent(1): the agent will now delete all loaded keys when signaled with SIGUSR1. This allows deletion of keys without having access to $SSH_AUTH_SOCK.

* Portable OpenSSH, ssh-agent(1): support systemd-style socket activation in ssh-agent using the LISTEN_PID/LISTEN_FDS mechanism. Activated when these environment variables are set, the agent is started with the -d or -D option and no socket path is set. GHPR502

* ssh-keygen(1): support FIDO tokens that return no attestation data, e.g. recent WinHello. GHPR542

* ssh-agent(1): add a "-Owebsafe-allow=..." option to allow the

default FIDO application ID allow-list to be overridden.

* Add a work-in-progress tool to verify FIDO attestation blobs that ssh-keygen can optionally write when enrolling FIDO keys. This tool is available under regress/misc/ssh-verify-attestation for experimentation but is not installed by "make install".

* ssh-keygen(1): allow "-" as output file for moduli screening. GHPR393

Bugfixes
----

* sshd(8): remove assumption that the sshd_config and any configs it includes can fit in a (possibly enlarged) socket buffer. Previously it was possible to create a sufficiently large configuration that could cause sshd to fail to accept any connection. sshd(8) will now actively manage sending its config to the sshd-session sub-process.

* ssh(1): don't start the ObscureKeystrokeTiming mitigations if there has been traffic on a X11 forwarding channel recently. Should fix X11 forwarding performance problems when this setting is enabled. bz3655

* ssh(1): prohibit the comma character in hostnames accepted, but allow an underscore as the first character in a hostname.

* sftp(1): set high-water when resuming a "put". Prevents bogus "server reordered acks" debug message.

* ssh(1), sshd(8): fix regression in openssh-9.8, which would fail to accept "Match criteria=argument" as well as the documented "Match criteria argument" syntax in ssh_config and sshd_config. bz3739

* sftp(1), ssh(1): fix a number possible NULL dereference bugs, including Coverity CIDs 405019 and 477813.

* sshd(8): fix PerSourcePenalty incorrectly using "crash" penalty when LoginGraceTime was exceeded. bz3797

* sshd(8): fix "Match invalid-user" from incorrectly being activated in initial configuration pass when no other predicates were present on the match line

* sshd(8): fix debug logging of user specific delay. GHPR#552

* sshd(8): improve debug logging across sub-process boundaries. Previously some log messages were lost early in the sshd-auth and sshd-session processes' life.

* ssh(1): require control-escape character sequences passed via the '-e ^x' command-line to be exactly two characters long. Avoids one byte out-of-bounds read if ssh is invoked as "ssh -e^ ..." GHPR368

* ssh(1), sshd(8): prevent integer overflow in x11 port handling. These are theoretically possible if the admin misconfigured X11DisplayOffset or the user misconfigures their own $DISPLAY, but don't happen in normal operation. bz#3730

* ssh-keygen(1): don't mess up ssh-keygen -l output when the file contains CR characters; GHPR236 bz3385.

* sshd(8): add rate limits to logging of connections dropped by PerSourcePenalties. Previously these could be noisy in logs.

* ssh(1): fix argument of "Compression" directive in ssh -G config dump, which regressed in openssh-9.8.

* sshd(8): fix a corner-case triggered by UpdateHostKeys when sshd refuses to accept the signature returned by an agent holding host keys during the hostkey rotation sub-protocol. This situation could occur in situations where a PKCS#11 smartcard that lacked support for particular signature algorithms was used to store host keys.

* ssh-keygen(1): when using RSA keys to sign messages with "ssh-keygen -Y", select the signature algorithm based on the requested hash algorithm ("-Ohashalg=xxx"). This allows using something other than the default of rsa-sha2-512, which may not be supported on all signing backends, e.g. some smartcards only support SHA256.

* ssh(1), sshd(8), ssh-keyscan(1): fix ML-KEM768x25519 KEX on big-endian systems.

* Many regression and interop test improvements.

Portability
----

* All: add support for AWS-LC (AWS libcrypto). bz3784

* sshd(8): add wtmpdb support as a Y2038 safe wtmp replacement.

* sshd(8): add support for locking sshd into memory, enabled with

```
    the --with-linux-memlock-onfault configure flag.

 * Add support for building a standalone sk-libfido2 library,
   enabled by --with-security-key-standalone

 * ssh(1), sshd(8), ssh-keyscan(1): include __builtin_popcount
   replacement function. for compilers that lack it.

 * All: Check for and replace le32toh, le64toh, htole64 separately.
   It appears that at least some versions of endian.h in glibc do
   not have the latter two. bz#3794

 * Remove ancient RHL 6.x config in RPM spec.

Checksums:
==========

 - SHA1 (openssh-10.0.tar.gz) = 933f4fded0497ef6a588381257276e156a70f9c3
 - SHA256 (openssh-10.0.tar.gz) = oaJ+cXLCVoCZAz9W5W1vF7ko4GJW6iq7JmblrPUA34Q=

 - SHA1 (openssh-10.0p2.tar.gz) = ac4205e827aea383bf316a33a0e2d5b66b85fcf8
 - SHA256 (openssh-10.0p2.tar.gz) = AhoucJoO30JQsSVr1anlAEEakN3avqgw7VnO+Q652Fw=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available from the mirror sites:
https://cdn.openbsd.org/pub/OpenBSD/OpenSSH/RELEASE_KEY.asc

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com
```

OpenSSH 9.9p2 (2025-02-18)

```
OpenSSH 9.9p2 was released on 2025-02-18. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
https://www.openssh.com/donations.html

Changes since OpenSSH 9.9p1
===========================

This release fixes two security bugs.

Security
========

* Fix CVE-2025-26465 - ssh(1) in OpenSSH versions 6.8p1 to 9.9p1
  (inclusive) contained a logic error that allowed an on-path
  attacker (a.k.a MITM) to impersonate any server when the
  VerifyHostKeyDNS option is enabled. This option is off by default.

* Fix CVE-2025-26466 - sshd(8) in OpenSSH versions 9.5p1 to 9.9p1
  (inclusive) is vulnerable to a memory/CPU denial-of-service related
  to the handling of SSH2_MSG_PING packets. This condition may be
  mitigated using the existing PerSourcePenalties feature.

Both vulnerabilities were discovered and demonstrated to be exploitable
by the Qualys Security Advisory team. We thank them for their detailed
review of OpenSSH.

For OpenBSD, fixes to these problems are available as errata; refer
to https://www.openbsd.org/errata.html

Bugfixes
========

 * ssh(1), sshd(8): fix regression in Match directive that caused
   failures when predicates and their arguments were separated by '='
   characters instead of whitespace (bz3739).

 * sshd(8): fix the "Match invalid-user" predicate, which was matching
   incorrectly in the initial pass of config evaluation.

 * ssh(1), sshd(8), ssh-keyscan(1): fix mlkem768x25519-sha256 key
   exchange on big-endian systems.

 * Fix a number of build problems on particular operating systems /
   configurations.

Checksums:
==========

 - SHA1 (openssh-9.9p2.tar.gz) = edefe960645780dee78059c444d4261667ad3056
 - SHA256 (openssh-9.9p2.tar.gz) = karbYD4IzChe3fll4RmdAlhfqU2ZTWyuW0Hhch4hVnM=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
```

key used to sign the releases is available from the mirror sites:
https://cdn.openbsd.org/pub/OpenBSD/OpenSSH/RELEASE_KEY.asc

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

---

OpenSSH 9.9/9.9p1 (2024-09-19)

OpenSSH 9.9 was released on 2024-09-19. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
https://www.openssh.com/donations.html

Future deprecation notice
=========================

OpenSSH plans to remove support for the DSA signature algorithm in
early 2025. This release disables DSA by default at compile time.

DSA, as specified in the SSHv2 protocol, is inherently weak - being
limited to a 160 bit private key and use of the SHA1 digest. Its
estimated security level is only 80 bits symmetric equivalent.

OpenSSH has disabled DSA keys by default since 2015 but has retained
run-time optional support for them. DSA was the only mandatory-to-
implement algorithm in the SSHv2 RFCs, mostly because alternative
algorithms were encumbered by patents when the SSHv2 protocol was
specified.

This has not been the case for decades at this point and better
algorithms are well supported by all actively-maintained SSH
implementations. We do not consider the costs of maintaining DSA
in OpenSSH to be justified and hope that removing it from OpenSSH
can accelerate its wider deprecation in supporting cryptography
libraries.

Currently DSA is disabled at compile time. The final step of
removing DSA support entirely is planned for the first OpenSSH
release of 2025.

DSA support may be re-enabled on OpenBSD by setting "DSAKEY=yes"
in Makefile.inc. To enable DSA support in portable OpenSSH, pass
the "--enable-dsa-keys" option to configure.

Potentially-incompatible changes
----

 * ssh(1): remove support for pre-authentication compression.
   OpenSSH has only supported post-authentication compression in
   the server for some years. Compression before authentication
   significantly increases the attack surface of SSH servers and risks
   creating oracles that reveal information about information sent
   during authentication.

 * ssh(1), sshd(8): processing of the arguments to the "Match"
   configuration directive now follows more shell-like rules for
   quoted strings, including allowing nested quotes and \-escaped
   characters. If configurations contained workarounds for the
   previous simplistic quote handling then they may need to be
   adjusted. If this is the case, it's most likely to be in the
   arguments to a "Match exec" confition. In this case, moving the
   command to be evaluated from the Match line to an external shell
   script is easiest way to preserve compatibility with both the old
   and new versions.

Changes since OpenSSH 9.8
=========================

This release contains a number of new features and bugfixes.

New features
----

 * ssh(1), sshd(8): add support for a new hybrid post-quantum key
   exchange based on the FIPS 203 Module-Lattice Key Enapsulation
   mechanism (ML-KEM) combined with X25519 ECDH as described by
   https://datatracker.ietf.org/doc/html/draft-kampanakis-curdle-ssh-pq-ke-03
   This algorithm "mlkem768x25519-sha256" is available by default.

 * ssh(1): the ssh_config "Include" directive can now expand
   environment as well as the same set of %-tokens "Match Exec"
   supports.

 * sshd(8): add a sshd_config "RefuseConnection" option that, if set
   will terminate the connection at the first authentication request.

 * sshd(8): add a "refuseconnection" penalty class to sshd_config
   PerSourcePenalties that is applied when a connection is dropped by
   the new RefuseConnection keyword.

 * sshd(8): add a "Match invalid-user" predicate to sshd_config Match
   options that matches when the target username is not valid on the
   server.

 * ssh(1), sshd(8): update the Streamlined NTRUPrime code to a
   substantially faster implementation.

 * ssh(1), sshd(8): the hybrid Streamlined NTRUPrime/X25519 key
   exchange algorithm now has an IANA-assigned name in addition to
   the "@openssh.com" vendor extension name. This algorithm is now
   also available under this name "sntrup761x25519-sha512"

 * ssh(1), sshd(8), ssh-agent(1): prevent private keys from being
   included in core dump files for most of their lifespans. This is
   in addition to pre-existing controls in ssh-agent(1) and sshd(8)
   that prevented coredumps. This feature is supported on OpenBSD,
   Linux and FreeBSD.

 * All: convert key handling to use the libcrypto EVP_PKEY API, with
   the exception of DSA.

 * sshd(8): add a random amount of jitter (up to 4 seconds) to the
   grace login time to make its expiry unpredictable.

Bugfixes
----

 * sshd(8): relax absolute path requirement back to what it was prior
   to OpenSSH 9.8, which incorrectly required that sshd was started
   with an absolute path in inetd mode. bz3717

 * sshd(8): fix regression introduced in openssh-9.8 that swapped the
   order of source and destination addresses in some sshd log messages.

 * sshd(8): do not apply authorized_keys options when signature
   verification fails. Prevents more restrictive key options being
   incorrectly applied to subsequent keys in authorized_keys. bz3733

 * ssh-keygen(1): include pathname in some of ssh-keygen's passphrase
   prompts. Helps the user know what's going on when ssh-keygen is
   invoked via other tools. Requested in GHPR503

 * ssh(1), ssh-add(1): make parsing user@host consistently look for
   the last '@' in the string rather than the first. This makes it
   possible to more consistently use usernames that contain '@'
   characters.

 * ssh(1), sshd(8): be more strict in parsing key type names. Only
   allow short names (e.g "rsa") in user-interface code and require
   full SSH protocol names (e.g. "ssh-rsa") everywhere else. bz3725

 * regress: many performance and correctness improvements to the
   re-keying regression test.

 * ssh-keygen(1): clarify that ed25519 is the default key type
   generated and clarify that rsa-sha2-512 is the default signature
   scheme when RSA is in use. GHPR505

 * sshd(8): fix minor memory leak in Subsystem option parsing; GHPR515

 * All: additional hardening and consistency checks for the sshbuf
   code.

 * sshd(8): reduce default logingrace penalty to ensure that a single
   forgotton login that times out will be below the penalty threshold.

 * ssh(1): fix proxy multiplexing (-O proxy) bug. If a mux started with
   ControlPersist then later has a forwarding added using mux proxy
   connection and the forwarding was used, then when the mux proxy
   session terminated, the mux master process would issue a bad message
   that terminated the connection.

Portability
----

 * sync contrib/ssh-copy-id to the latest upstream version.

 * regress: improve portablility for some awk(1) usage (e.g. Solaris)

 * In the contrib/redhat RPM spec file, without_openssl was previously
   incorrectly enabled unconditionally.

 * sshd(8) restore audit call before exit that regressed in openssh-9.8
   Fixes an issue where the SSH_CONNECTION_ABANDON event was not
   recorded.

 * sshd(8): add support for class-imposed loging restrictions on FreeBSD.
   Allowing auth_hostok(3) and auth_timeok(3) to control logins.

 * Build fixes for Musl libc.

 * Fix detection of setres*id on GNU/Hurd

```
Checksums:
==========

 - SHA1 (openssh-9.9.tar.gz) = 080acf6ff0b862e8faa3baa3920a079536d28e85
 - SHA256 (openssh-9.9.tar.gz) = h1xwa7CVcJfN7I9MgxxPBpUELzo+tnmLy+6slYHTUtw=

 - SHA1 (openssh-9.9p1.tar.gz) = 5ded7eb0add0b02b5d1a1c4bf5cb2c89d2117b53
 - SHA256 (openssh-9.9p1.tar.gz) = s0P7zb/4fxWxmG5uFdbU/Jp9NgZr5rf7UHCHuo+WbAI=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available from the mirror sites:
https://cdn.openbsd.org/pub/OpenBSD/OpenSSH/RELEASE_KEY.asc

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com
```

OpenSSH 9.8/9.8p1 (2024-07-01)

```
OpenSSH 9.8 was released on 2024-07-01. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
https://www.openssh.com/donations.html

Security
========

This release contains fixes for two security problems, one critical
and one minor.

1) Race condition in sshd(8)

A critical vulnerability in sshd(8) was present in Portable OpenSSH
versions between 8.5p1 and 9.7p1 (inclusive) that may allow arbitrary
code execution with root privileges.

Successful exploitation has been demonstrated on 32-bit Linux/glibc
systems with ASLR. Under lab conditions, the attack requires on
average 6-8 hours of continuous connections up to the maximum the
server will accept. Exploitation on 64-bit systems is believed to be
possible but has not been demonstrated at this time. It's likely that
these attacks will be improved upon.

Exploitation on non-glibc systems is conceivable but has not been
examined. Systems that lack ASLR or users of downstream Linux
distributions that have modified OpenSSH to disable per-connection
ASLR re-randomisation (yes - this is a thing, no - we don't
understand why) may potentially have an easier path to exploitation.
OpenBSD is not vulnerable.

We thank the Qualys Security Advisory Team for discovering, reporting
and demonstrating exploitability of this problem, and for providing
detailed feedback on additional mitigation measures.

2) Logic error in ssh(1) ObscureKeystrokeTiming

In OpenSSH version 9.5 through 9.7 (inclusive), when connected to an
OpenSSH server version 9.5 or later, a logic error in the ssh(1)
ObscureKeystrokeTiming feature (on by default) rendered this feature
ineffective - a passive observer could still detect which network
packets contained real keystrokes when the countermeasure was active
because both fake and real keystroke packets were being sent
unconditionally.

This bug was found by Philippos Giavridis and also independently by
Jacky Wei En Kung, Daniel Hugenroth and Alastair Beresford of the
University of Cambridge Computer Lab.

Worse, the unconditional sending of both fake and real keystroke
packets broke another long-standing timing attack mitigation. Since
OpenSSH 2.9.9 sshd(8) has sent fake keystoke echo packets for
traffic received on TTYs in echo-off mode, such as when entering a
password into su(8) or sudo(8). This bug rendered these fake
keystroke echoes ineffective and could allow a passive observer of
a SSH session to once again detect when echo was off and obtain
fairly limited timing information about keystrokes in this situation
(20ms granularity by default).

This additional implication of the bug was identified by Jacky Wei
En Kung, Daniel Hugenroth and Alastair Beresford and we thank them
for their detailed analysis.

This bug does not affect connections when ObscureKeystrokeTiming
was disabled or sessions where no TTY was requested.

Future deprecation notice
```

```
========================

OpenSSH plans to remove support for the DSA signature algorithm in
early 2025. This release disables DSA by default at compile time.

DSA, as specified in the SSHv2 protocol, is inherently weak - being
limited to a 160 bit private key and use of the SHA1 digest. Its
estimated security level is only 80 bits symmetric equivalent.

OpenSSH has disabled DSA keys by default since 2015 but has retained
run-time optional support for them. DSA was the only mandatory-to-
implement algorithm in the SSHv2 RFCs, mostly because alternative
algorithms were encumbered by patents when the SSHv2 protocol was
specified.

This has not been the case for decades at this point and better
algorithms are well supported by all actively-maintained SSH
implementations. We do not consider the costs of maintaining DSA
in OpenSSH to be justified and hope that removing it from OpenSSH
can accelerate its wider deprecation in supporting cryptography
libraries.

This release, and its deactivation of DSA by default at compile-time,
marks the second step in our timeline to finally deprecate DSA. The
final step of removing DSA support entirely is planned for the first
OpenSSH release of 2025.

DSA support may be re-enabled in OpenBSD by setting "DSAKEY=yes"
in Makefile.inc. To enable DSA support in portable OpenSSH, pass
the "--enable-dsa-keys" option to configure.

Potentially-incompatible changes
--------------------------------

 * all: as mentioned above, the DSA signature algorithm is now
   disabled at compile time.

 * sshd(8): the server will now block client addresses that
   repeatedly fail authentication, repeatedly connect without ever
   completing authentication or that crash the server. See the
   discussion of PerSourcePenalties below for more information.
   Operators of servers that accept connections from many users, or
   servers that accept connections from addresses behind NAT or
   proxies may need to consider these settings.

 * sshd(8): the server has been split into a listener binary, sshd(8),
   and a per-session binary "sshd-session". This allows for a much
   smaller listener binary, as it no longer needs to support the SSH
   protocol. As part of this work, support for disabling privilege
   separation (which previously required code changes to disable) and
   disabling re-execution of sshd(8) has been removed. Further
   separation of sshd-session into additional, minimal binaries is
   planned for the future.

 * sshd(8): several log messages have changed. In particular, some
   log messages will be tagged with as originating from a process
   named "sshd-session" rather than "sshd".

 * ssh-keyscan(1): this tool previously emitted comment lines
   containing the hostname and SSH protocol banner to standard error.
   This release now emits them to standard output, but adds a new
   "-q" flag to silence them altogether.

 * sshd(8): (portable OpenSSH only) sshd will no longer use argv[0]
   as the PAM service name. A new "PAMServiceName" sshd_config(5)
   directive allows selecting the service name at runtime. This
   defaults to "sshd". bz2101

 * (portable OpenSSH only) Automatically-generated files, such as
   configure, config.h.in, etc will now be checked in to the portable
   OpenSSH git release branch (e.g. V_9_8). This should ensure that
   the contents of the signed release branch exactly match the
   contents of the signed release tarball.

Changes since OpenSSH 9.7
=========================

This release contains mostly bugfixes.

New features
------------

 * sshd(8): as described above, sshd(8) will now penalise client
   addresses that, for various reasons, do not successfully complete
   authentication. This feature is controlled by a new sshd_config(5)
   PerSourcePenalties option and is on by default.

   sshd(8) will now identify situations where the session did not
   authenticate as expected. These conditions include when the client
   repeatedly attempted authentication unsucessfully (possibly
   indicating an attack against one or more accounts, e.g. password
   guessing), or when client behaviour caused sshd to crash (possibly
   indicating attempts to exploit bugs in sshd).

   When such a condition is observed, sshd will record a penalty of
   some duration (e.g. 30 seconds) against the client's address. If
   this time is above a minimum configurable threshold, then all
   connections from the client address will be refused (along with any
```

```
    others in the same PerSourceNetBlockSize CIDR range) until the
    penalty expire.

    Repeated offenses by the same client address will accrue greater
    penalties, up to a configurable maximum. Address ranges may be
    fully exempted from penalties, e.g. to guarantee access from a set
    of trusted management addresses, using the new sshd_config(5)
    PerSourcePenaltyExemptList option.

    We hope these options will make it significantly more difficult for
    attackers to find accounts with weak/guessable passwords or exploit
    bugs in sshd(8) itself. This option is enabled by default.

  * ssh(8): allow the HostkeyAlgorithms directive to disable the
    implicit fallback from certificate host key to plain host keys.

Bugfixes
--------

  * misc: fix a number of inaccuracies in the PROTOCOL.*
    documentation files. GHPR430 GHPR487

  * all: switch to strtonum(3) for more robust integer parsing in most
    places.

  * ssh(1), sshd(8): correctly restore sigprocmask around ppoll()

  * ssh-keysign(8): stricter validation of messaging socket fd GHPR492

  * sftp(1): flush stdout after writing "sftp>" prompt when not using
    editline. GHPR480

  * sftp-server(8): fix home-directory extension implementation, it
    previously always returned the current user's home directory
    contrary to the spec. GHPR477

  * ssh-keyscan(1): do not close stdin to prevent error messages when
    stdin is read multiple times. E.g.
    echo localhost | ssh-keyscan -f - -f -

  * regression tests: fix rekey test that was testing the same KEX
    algorithm repeatedly instead of testing all of them. bz3692

  * ssh_config(5), sshd_config(5): clarify the KEXAlgorithms directive
    documentation, especially around what is supported vs available.
    bz3701.

Portability
-----------

  * sshd(8): expose SSH_AUTH_INFO_0 always to PAM auth modules
    unconditionally. The previous behaviour was to expose it only when
    particular authentication methods were in use.

  * build: fix OpenSSL ED25519 support detection. An incorrect function
    signature in configure.ac previously prevented enabling the recently
    added support for ED25519 private keys in PEM PKCS8 format.

  * ssh(1), ssh-agent(8): allow the presence of the WAYLAND_DISPLAY
    environment variable to enable SSH_ASKPASS, similarly to the X11
    DISPLAY environment variable. GHPR479

  * build: improve detection of the -fzero-call-used-regs compiler
    flag. bz3673.

  * build: relax OpenSSL version check to accept all OpenSSL 3.x
    versions.

  * sshd(8): add support for notifying systemd on server listen and
    reload, using a standalone implementation that doesn't depend on
    libsystemd. bz2641

Checksums:
==========

  - SHA1 (openssh-9.8.tar.gz) = bc45cedae7f70b41e9922ef4c9f56e74b9a659b7
  - SHA256 (openssh-9.8.tar.gz) = Dnc69VLWFBFdiaz8wySPlvHjb7wZfh/kblQ8ISuQr1Y=

  - SHA1 (openssh-9.8p1.tar.gz) = a0bb501b11349f5c5c33a269351be091dc2c2727
  - SHA256 (openssh-9.8p1.tar.gz) = 3YvQAqN5tdSZ37BQ3R+pr4Ap6ARh9LtsUjxJlz9aOfM=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available from the mirror sites:
https://cdn.openbsd.org/pub/OpenBSD/OpenSSH/RELEASE_KEY.asc

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com
```

OpenSSH 9.7/9.7p1 (2024-03-11)

OpenSSH 9.7 was released on 2024-03-11. It is available from the mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their continued support of the project, especially those who contributed code or patches, reported bugs, tested snapshots or donated to the project. More information on donations may be found at: https://www.openssh.com/donations.html

Future deprecation notice
==========

OpenSSH plans to remove support for the DSA signature algorithm in early 2025 and compile-time disable it later this year.

DSA, as specified in the SSHv2 protocol, is inherently weak - being limited to a 160 bit private key and use of the SHA1 digest. Its estimated security level is only 80 bits symmetric equivalent.

OpenSSH has disabled DSA keys by default since 2015 but has retained run-time optional support for them. DSA was the only mandatory-to-implement algorithm in the SSHv2 RFCs[3], mostly because alternative algorithms were encumbered by patents when the SSHv2 protocol was specified.

This has not been the case for decades at this point and better algorithms are well supported by all actively-maintained SSH implementations. We do not consider the costs of maintaining DSA in OpenSSH to be justified and hope that removing it from OpenSSH can accelerate its wider deprecation in supporting cryptography libraries.

This release makes DSA support in OpenSSH compile-time optional, defaulting to on. We intend the next release to change the default to disable DSA at compile time. The first OpenSSH release of 2025 will remove DSA support entirely.

Changes since OpenSSH 9.6
==========

This release contains mostly bugfixes.

New features
---

 * ssh(1), sshd(8): add a "global" ChannelTimeout type that watches all open channels and will close all open channels if there is no traffic on any of them for the specified interval. This is in addition to the existing per-channel timeouts added recently.

   This supports situations like having both session and x11 forwarding channels open where one may be idle for an extended period but the other is actively used. The global timeout could close both channels when both have been idle for too long.

 * All: make DSA key support compile-time optional, defaulting to on.

Bugfixes
---

 * sshd(8): don't append an unnecessary space to the end of subsystem arguments (bz3667)

 * ssh(1): fix the multiplexing "channel proxy" mode, broken when keystroke timing obfuscation was added. (GHPR#463)

 * ssh(1), sshd(8): fix spurious configuration parsing errors when options that accept array arguments are overridden (bz3657).

 * ssh-agent(1): fix potential spin in signal handler (bz3670)

 * Many fixes to manual pages and other documentation, including GHPR#462, GHPR#454, GHPR#442 and GHPR#441.

 * Greatly improve interop testing against PuTTY.

Portability
---

 * Improve the error message when the autoconf OpenSSL header check fails (bz#3668)

 * Improve detection of broken toolchain -fzero-call-used-regs support (bz3645).

 * Fix regress/misc/fuzz-harness fuzzers and make them compile without warnings when using clang16

Checksums:
==========

 - SHA1 (openssh-9.7.tar.gz) = 163272058edc20a8fde81661734a6684c9b4db11
 - SHA256 (openssh-9.7.tar.gz) = gXDWrF4wN2UWyPjyjvVhpjjKd7D2qI6LyZiIYhbJQVg=

 - SHA1 (openssh-9.7p1.tar.gz) = ce8985ea0ea2f16a5917fd982ade0972848373cc
 - SHA256 (openssh-9.7p1.tar.gz) = SQQm92bYKidj/KzY2D6j1weYdQx70q/y5X3FZg93P/0=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available from the mirror sites:
https://cdn.openbsd.org/pub/OpenBSD/OpenSSH/RELEASE_KEY.asc

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

---

OpenSSH 9.6/9.6p1 (2023-12-18)

OpenSSH 9.6 was released on 2023-12-18. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
https://www.openssh.com/donations.html

Changes since OpenSSH 9.5
=========================

This release contains a number of security fixes, some small features
and bugfixes.

Security
========

This release contains fixes for a newly-discovered weakness in the
SSH transport protocol, a logic error relating to constrained PKCS#11
keys in ssh-agent(1) and countermeasures for programs that invoke
ssh(1) with user or hostnames containing invalid characters.

 * ssh(1), sshd(8): implement protocol extensions to thwart the
   so-called "Terrapin attack" discovered by Fabian Bäumer, Marcus
   Brinkmann and Jörg Schwenk. This attack allows a MITM to effect a
   limited break of the integrity of the early encrypted SSH transport
   protocol by sending extra messages prior to the commencement of
   encryption, and deleting an equal number of consecutive messages
   immediately after encryption starts. A peer SSH client/server
   would not be able to detect that messages were deleted.

   While cryptographically novel, the security impact of this attack
   is fortunately very limited as it only allows deletion of
   consecutive messages, and deleting most messages at this stage of
   the protocol prevents user user authentication from proceeding and
   results in a stuck connection.

   The most serious identified impact is that it lets a MITM to
   delete the SSH2_MSG_EXT_INFO message sent before authentication
   starts, allowing the attacker to disable a subset of the keystroke
   timing obfuscation features introduced in OpenSSH 9.5. There is no
   other discernable impact to session secrecy or session integrity.

   OpenSSH 9.6 addresses this protocol weakness through a new "strict
   KEX" protocol extension that will be automatically enabled when
   both the client and server support it. This extension makes
   two changes to the SSH transport protocol to improve the integrity
   of the initial key exchange.

   Firstly, it requires endpoints to terminate the connection if any
   unnecessary or unexpected message is received during key exchange
   (including messages that were previously legal but not strictly
   required like SSH2_MSG_DEBUG). This removes most malleability from
   the early protocol.

   Secondly, it resets the Message Authentication Code counter at the
   conclusion of each key exchange, preventing previously inserted
   messages from being able to make persistent changes to the
   sequence number across completion of a key exchange. Either of
   these changes should be sufficient to thwart the Terrapin Attack.

   More details of these changes are in the PROTOCOL file in the
   OpenSSH source distribition.

 * ssh-agent(1): when adding PKCS#11-hosted private keys while
   specifying destination constraints, if the PKCS#11 token returned
   multiple keys then only the first key had the constraints applied.
   Use of regular private keys, FIDO tokens and unconstrained keys
   are unaffected.

 * ssh(1): if an invalid user or hostname that contained shell
   metacharacters was passed to ssh(1), and a ProxyCommand,
   LocalCommand directive or "match exec" predicate referenced the
   user or hostname via %u, %h or similar expansion token, then
   an attacker who could supply arbitrary user/hostnames to ssh(1)
   could potentially perform command injection depending on what
   quoting was present in the user-supplied ssh_config(5) directive.

   This situation could arise in the case of git submodules, where

a repository could contain a submodule with shell characters in its user/hostname. Git does not ban shell metacharacters in user or host names when checking out repositories from untrusted sources.

Although we believe it is the user's responsibility to ensure validity of arguments passed to ssh(1), especially across a security boundary such as the git example above, OpenSSH 9.6 now bans most shell metacharacters from user and hostnames supplied via the command-line. This countermeasure is not guaranteed to be effective in all situations, as it is infeasible for ssh(1) to universally filter shell metacharacters potentially relevant to user-supplied commands.

User/hostnames provided via ssh_config(5) are not subject to these restrictions, allowing configurations that use strange names to continue to be used, under the assumption that the user knows what they are doing in their own configuration files.

Potentially incompatible changes
---

* ssh(1), sshd(8): the RFC4254 connection/channels protocol provides a TCP-like window mechanism that limits the amount of data that can be sent without acceptance from the peer. In cases where this limit was exceeded by a non-conforming peer SSH implementation, ssh(1)/sshd(8) previously discarded the extra data. From OpenSSH 9.6, ssh(1)/sshd(8) will now terminate the connection if a peer exceeds the window limit by more than a small grace factor. This change should have no effect of SSH implementations that follow the specification.

New features
---

* ssh(1): add a %j token that expands to the configured ProxyJump hostname (or the empty string if this option is not being used) that can be used in a number of ssh_config(5) keywords. bz3610

* ssh(1): add ChannelTimeout support to the client, mirroring the same option in the server and allowing ssh(1) to terminate quiescent channels.

* ssh(1), sshd(8), ssh-add(1), ssh-keygen(1): add support for reading ED25519 private keys in PEM PKCS8 format. Previously only the OpenSSH private key format was supported.

* ssh(1), sshd(8): introduce a protocol extension to allow renegotiation of acceptable signature algorithms for public key authentication after the server has learned the username being used for authentication. This allows varying sshd_config(5) PubkeyAcceptedAlgorithms in a "Match user" block.

* ssh-add(1), ssh-agent(1): add an agent protocol extension to allow specifying certificates when loading PKCS#11 keys. This allows the use of certificates backed by PKCS#11 private keys in all OpenSSH tools that support ssh-agent(1). Previously only ssh(1) supported this use-case.

Bugfixes
---

* ssh(1): when deciding whether to enable the keystroke timing obfuscation, enable it only if a channel with a TTY is active.

* ssh(1): switch mainloop from poll(3) to ppoll(3) and mask signals before checking flags set in signal handler. Avoids potential race condition between signaling ssh to exit and polling. bz3531

* ssh(1): when connecting to a destination with both the AddressFamily and CanonicalizeHostname directives in use, the AddressFamily directive could be ignored. bz5326

* sftp(1): correct handling of the limits@openssh.com option when the server returned an unexpected message.

* A number of fixes to the PuTTY and Dropbear regress/integration tests.

* ssh(1): release GSS OIDs only at end of authentication, avoiding unnecessary init/cleanup cycles. bz2982

* ssh_config(5): mention "none" is a valid argument to IdentityFile in the manual. bz3080

* scp(1): improved debugging for paths from the server rejected for not matching the client's glob(3) pattern in old SCP/RCP protocol mode.

* ssh-agent(1): refuse signing operations on destination-constrained keys if a previous session-bind operation has failed. This may prevent a fail-open situation in future if a user uses a mismatched ssh(1) client and ssh-agent(1) where the client supports a key type that the agent does not support.

Portability

```
 * Better identify unsupported and unstable compiler flags, such as
   -fzero-call-used-regs which has been unstable across a several
   clang releases.

 * A number of fixes to regression test reliability and log
   collection.

 * Update the OpenSSL dependency in the RPM specification.

 * sshd(8): for OpenSolaris systems that support privilege limitation
   via the getpflags() interface, prefer using the newer PRIV_XPOLICY
   to PRIV_LIMIT. bz2833

Checksums:
==========

 - SHA1 (openssh-9.6.tar.gz) = a6d4cb69811e879e2f158c2e597fd9f444b26506
 - SHA256 (openssh-9.6.tar.gz) = nejPUhSnG1R1sOmIBi/t+HMNvsRqfN/DJgjwIU2tvqg=

 - SHA1 (openssh-9.6p1.tar.gz) = de300d09ec79fdbf37de4e6672cce4161439f2c3
 - SHA256 (openssh-9.6p1.tar.gz) = kQIRwHJVqMWtZUORtA7lmABxDdgRndU2LeCThap6d3w=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available from the mirror sites:
https://cdn.openbsd.org/pub/OpenBSD/OpenSSH/RELEASE_KEY.asc

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com
```

OpenSSH 9.5/9.5p1 (2023-10-04)

```
OpenSSH 9.5 was released on 2023-10-04. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
https://www.openssh.com/donations.html

Changes since OpenSSH 9.4
=========================

This release fixes a number of bugs and adds some small features.

Potentially incompatible changes
--------------------------------

 * ssh-keygen(1): generate Ed25519 keys by default. Ed25519 public keys
   are very convenient due to their small size. Ed25519 keys are
   specified in RFC 8709 and OpenSSH has supported them since version 6.5
   (January 2014).

 * sshd(8): the Subsystem directive now accurately preserves quoting of
   subsystem commands and arguments. This may change behaviour for exotic
   configurations, but the most common subsystem configuration
   (sftp-server) is unlikely to be affected.

New features
------------

 * ssh(1): add keystroke timing obfuscation to the client. This attempts
   to hide inter-keystroke timings by sending interactive traffic at
   fixed intervals (default: every 20ms) when there is only a small
   amount of data being sent. It also sends fake "chaff" keystrokes for
   a random interval after the last real keystroke. These are
   controlled by a new ssh_config ObscureKeystrokeTiming keyword.

 * ssh(1), sshd(8): Introduce a transport-level ping facility. This adds
   a pair of SSH transport protocol messages SSH2_MSG_PING/PONG to
   implement a ping capability. These messages use numbers in the "local
   extensions" number space and are advertised using a "ping@openssh.com"
   ext-info message with a string version number of "0".

 * sshd(8): allow override of Subsystem directives in sshd Match blocks.

Bugfixes
--------

 * scp(1): fix scp in SFTP mode recursive upload and download of
   directories that contain symlinks to other directories. In scp mode,
   the links would be followed, but in SFTP mode they were not. bz3611

 * ssh-keygen(1): handle cr+lf (instead of just cr) line endings in
   sshsig signature files.

 * ssh(1): interactive mode for ControlPersist sessions if they
   originally requested a tty.
```

```
 * sshd(8): make PerSourceMaxStartups first-match-wins

 * sshd(8): limit artificial login delay to a reasonable maximum (5s)
   and don't delay at all for the "none" authentication mechanism.cw
    bz3602

 * sshd(8): Log errors in kex_exchange_identification() with level
   verbose instead of error to reduce preauth log spam. All of those
   get logged with a more generic error message by sshpkt_fatal().

 * sshd(8): correct math for ClientAliveInterval that caused the probes
    to be sent less frequently than configured.

 * ssh(1): fix regression in OpenSSH 9.4 (mux.c r1.99) that caused
   multiplexed sessions to ignore SIGINT under some circumstances.

Portability
-----------

 * Avoid clang zero-call-used-regs=all bug on Apple compilers, which
   for some reason have version numbers that do not match the upstream
   clang version numbers. bz#3584

 * Fix configure test for zlib 1.3 and later/development versions. bz3604

Checksums:
==========

 - SHA1 (openssh-9.5.tar.gz) = 8a0bd3a91fac338d97d91817af58df731f6509a3
 - SHA256 (openssh-9.5.tar.gz) = sVMxeM3d6g65qBMktJIofxmK4Ipg9dblKif0VnhPeO0=

 - SHA1 (openssh-9.5p1.tar.gz) = 35c16dcc6e7d0a9465faa241476ef24f76b196cc
 - SHA256 (openssh-9.5p1.tar.gz) = 8Cbnt5un+1QPdRgq+W3IqPHbOV+SK7yfbKYDZyaGCGs=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available from the mirror sites:
https://cdn.openbsd.org/pub/OpenBSD/OpenSSH/RELEASE_KEY.asc

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com
```

OpenSSH 9.4/9.4p1 (2023-08-10)

```
OpenSSH 9.4 was released on 2023-08-10. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
https://www.openssh.com/donations.html

Changes since OpenSSH 9.3p2
===========================

This release fixes a number of bugs and adds some small features.

Potentially incompatible changes
--------------------------------

 * This release removes support for older versions of libcrypto.
   OpenSSH now requires LibreSSL >= 3.1.0 or OpenSSL >= 1.1.1.
   Note that these versions are already deprecated by their upstream
   vendors.

 * ssh-agent(1): PKCS#11 modules must now be specified by their full
   paths. Previously dlopen(3) could search for them in system
   library directories.

New features
------------

 * ssh(1): allow forwarding Unix Domain sockets via ssh -W.

 * ssh(1): add support for configuration tags to ssh(1).
   This adds a ssh_config(5) "Tag" directive and corresponding
   "Match tag" predicate that may be used to select blocks of
   configuration similar to the pf.conf(5) keywords of the same
   name.

 * ssh(1): add a "match localnetwork" predicate. This allows matching
   on the addresses of available network interfaces and may be used to
   vary the effective client configuration based on network location.

 * ssh(1), sshd(8), ssh-keygen(1): infrastructure support for KRL
   extensions.  This defines wire formats for optional KRL extensions
   and implements parsing of the new submessages. No actual extensions
   are supported at this point.

 * sshd(8): AuthorizedPrincipalsCommand and AuthorizedKeysCommand now
```

```
    accept two additional %-expansion sequences: %D which expands to
    the routing domain of the connected session and %C which expands
    to the addresses and port numbers for the source and destination
    of the connection.

  * ssh-keygen(1): increase the default work factor (rounds) for the
    bcrypt KDF used to derive symmetric encryption keys for passphrase
    protected key files by 50%.

Bugfixes
--------

  * ssh-agent(1): improve isolation between loaded PKCS#11 modules
    by running separate ssh-pkcs11-helpers for each loaded provider.

  * ssh(1): make -f (fork after authentication) work correctly with
    multiplexed connections, including ControlPersist. bz3589 bz3589

  * ssh(1): make ConnectTimeout apply to multiplexing sockets and not
    just to network connections.

  * ssh-agent(1), ssh(1): improve defences against invalid PKCS#11
    modules being loaded by checking that the requested module
    contains the required symbol before loading it.

  * sshd(8): fix AuthorizedPrincipalsCommand when AuthorizedKeysCommand
    appears before it in sshd_config. Since OpenSSH 8.7 the
    AuthorizedPrincipalsCommand directive was incorrectly ignored in
    this situation. bz3574

  * sshd(8), ssh(1), ssh-keygen(1): remove vestigal support for KRL
    signatures When the KRL format was originally defined, it included
    support for signing of KRL objects. However, the code to sign KRLs
    and verify KRL signatues was never completed in OpenSSH. This
    release removes the partially-implemented code to verify KRLs.
    All OpenSSH tools now ignore KRL_SECTION_SIGNATURE sections in
    KRL files.

  * All: fix a number of memory leaks and unreachable/harmless integer
    overflows.

  * ssh-agent(1), ssh(1): don't truncate strings logged from PKCS#11
    modules; GHPR406

  * sshd(8), ssh(1): better validate CASignatureAlgorithms in
    ssh_config and sshd_config. Previously this directive would accept
    certificate algorithm names, but these were unusable in practice as
    OpenSSH does not support CA chains. bz3577

  * ssh(1): make `ssh -Q CASignatureAlgorithms` only list signature
    algorithms that are valid for CA signing. Previous behaviour was
    to list all signing algorithms, including certificate algorithms.

  * ssh-keyscan(1): gracefully handle systems where rlimits or the
    maximum number of open files is larger than INT_MAX; bz3581

  * ssh-keygen(1): fix "no comment" not showing on when running
    `ssh-keygen -l` on multiple keys where one has a comment and other
    following keys do not. bz3580

  * scp(1), sftp(1): adjust ftruncate() logic to handle servers that
    reorder requests. Previously, if the server reordered requests then
    the resultant file would be erroneously truncated.

  * ssh(1): don't incorrectly disable hostname canonicalization when
    CanonicalizeHostname=yes and ProxyJump was expicitly set to
    "none". bz3567

  * scp(1): when copying local->remote, check that the source file
    exists before opening an SFTP connection to the server. Based on
    GHPR#370

Portability
-----------

  * All: a number of build fixes for various platforms and
    configuration combinations.

  * sshd(8): provide a replacement for the SELinux matchpathcon()
    function, which is deprecated.

  * All: relax libcrypto version checks for OpenSSL >=3. Beyond
    OpenSSL 3.0, the ABI compatibility guarantees are wider (only
    the library major must match instead of major and minor in
    earlier versions).  bz#3548.

  * Tests: fix build problems for the sk-dummy.so FIDO provider module
    used in some tests.

Checksums:
==========

  - SHA1 (openssh-9.4.tar.gz) = d88126d8d7b8e5bf4656587ac4a16055560641cc
  - SHA256 (openssh-9.4.tar.gz) = 7eqFjx2hAunw+1Jy7f1JQXq//3AMr9B3dKtASDtq8go=

  - SHA1 (openssh-9.4p1.tar.gz) = 5dea1f3c88f9cfe53a711a3c893ee8b7d3ffecff
  - SHA256 (openssh-9.4p1.tar.gz) = Ngj9kIjbIWPOs+YAyFq3nQ3j0iHlkZLqGSPiMmOGaoU=
```

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available from the mirror sites:
https://cdn.openbsd.org/pub/OpenBSD/OpenSSH/RELEASE_KEY.asc

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

---

OpenSSH 9.3p2 (2023-07-19)

OpenSSH 9.3p2 was released on 2023-07-19. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
https://www.openssh.com/donations.html

Changes since OpenSSH 9.3
=========================

This release fixes a security bug.

Security
========

Fix CVE-2023-38408 - a condition where specific libaries loaded via
ssh-agent(1)'s PKCS#11 support could be abused to achieve remote
code execution via a forwarded agent socket if the following
conditions are met:

* Exploitation requires the presence of specific libraries on
  the victim system.
* Remote exploitation requires that the agent was forwarded
  to an attacker-controlled system.

Exploitation can also be prevented by starting ssh-agent(1) with an
empty PKCS#11/FIDO allowlist (ssh-agent -P '') or by configuring
an allowlist that contains only specific provider libraries.

This vulnerability was discovered and demonstrated to be exploitable
by the Qualys Security Advisory team.

In addition to removing the main precondition for exploitation,
this release removes the ability for remote ssh-agent(1) clients
to load PKCS#11 modules by default (see below).

Potentially-incompatible changes
----

 * ssh-agent(8): the agent will now refuse requests to load PKCS#11
   modules issued by remote clients by default. A flag has been added
   to restore the previous behaviour "-Oallow-remote-pkcs11".

   Note that ssh-agent(8) depends on the SSH client to identify
   requests that are remote. The OpenSSH >=8.9 ssh(1) client does
   this, but forwarding access to an agent socket using other tools
   may circumvent this restriction.

Checksums:
==========

- SHA1 (openssh-9.3p2.tar.gz) = 219cf700c317f400bb20b001c0406056f7188ea4
- SHA256 (openssh-9.3p2.tar.gz) = IA6+FH9ss/EB/QzfngJEKvfdyimN/9n0VoeOfMrGdug=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available from the mirror sites:
https://cdn.openbsd.org/pub/OpenBSD/OpenSSH/RELEASE_KEY.asc

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

---

OpenSSH 9.3/9.3p1 (2023-03-15)

OpenSSH 9.3 was released on 2023-03-15. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the

project. More information on donations may be found at:
https://www.openssh.com/donations.html

Changes since OpenSSH 9.2
=========================

This release fixes a number of security bugs.

Security
========

This release contains fixes for a security problem and a memory
safety problem. The memory safety problem is not believed to be
exploitable, but we report most network-reachable memory faults as
security bugs.

 * ssh-add(1): when adding smartcard keys to ssh-agent(1) with the
   per-hop destination constraints (ssh-add -h ...) added in OpenSSH
   8.9, a logic error prevented the constraints from being
   communicated to the agent. This resulted in the keys being added
   without constraints. The common cases of non-smartcard keys and
   keys without destination constraints are unaffected. This problem
   was reported by Luci Stanescu.

 * ssh(1): Portable OpenSSH provides an implementation of the
   getrrsetbyname(3) function if the standard library does not
   provide it, for use by the VerifyHostKeyDNS feature. A
   specifically crafted DNS response could cause this function to
   perform an out-of-bounds read of adjacent stack data, but this
   condition does not appear to be exploitable beyond denial-of-
   service to the ssh(1) client.

   The getrrsetbyname(3) replacement is only included if the system's
   standard library lacks this function and portable OpenSSH was not
   compiled with the ldns library (--with-ldns). getrrsetbyname(3) is
   only invoked if using VerifyHostKeyDNS to fetch SSHFP records. This
   problem was found by the Coverity static analyzer.

New features
----

 * ssh-keygen(1), ssh-keyscan(1): accept -Ohashalg=sha1|sha256 when
   outputting SSHFP fingerprints to allow algorithm selection. bz3493

 * sshd(8): add a `sshd -G` option that parses and prints the
   effective configuration without attempting to load private keys
   and perform other checks. This allows usage of the option before
   keys have been generated and for configuration evaluation and
   verification by unprivileged users.

Bugfixes
----

 * scp(1), sftp(1): fix progressmeter corruption on wide displays;
   bz3534

 * ssh-add(1), ssh-keygen(1): use RSA/SHA256 when testing usability
   of private keys as some systems are starting to disable RSA/SHA1
   in libcrypto.

 * sftp-server(8): fix a memory leak. GHPR363

 * ssh(1), sshd(8), ssh-keyscan(1): remove vestigal protocol
   compatibility code and simplify what's left.

 * Fix a number of low-impact Coverity static analysis findings.
   These include several reported via bz2687

 * ssh_config(5), sshd_config(5): mention that some options are not
   first-match-wins.

 * Rework logging for the regression tests. Regression tests will now
   capture separate logs for each ssh and sshd invocation in a test.

 * ssh(1): make `ssh -Q CASignatureAlgorithms` work as the manpage
   says it should; bz3532.

 * ssh(1): ensure that there is a terminating newline when adding a
   new entry to known_hosts; bz3529

Portability
----

 * sshd(8): harden Linux seccomp sandbox. Move to an allowlist of
   mmap(2), madvise(2) and futex(2) flags, removing some concerning
   kernel attack surface.

 * sshd(8): improve Linux seccomp-bpf sandbox for older systems;
   bz3537

Checksums:
==========

- SHA1 (openssh-9.3.tar.gz) = 5f9d2f73ddfe94f3f0a78bdf46704b6ad7b66ec7
- SHA256 (openssh-9.3.tar.gz) = eRcXkFZByz70DUBUcyIdvU0pVxP2X280FrmV8pyUdrk=

- SHA1 (openssh-9.3p1.tar.gz) = 610959871bf8d6baafc3525811948f85b5dd84ab
- SHA256 (openssh-9.3p1.tar.gz) = 6bq6dwGnalHz2Fpiw4OjydzZf6kAuFm8fbEUwYaK+Kg=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available from the mirror sites:
https://cdn.openbsd.org/pub/OpenBSD/OpenSSH/RELEASE_KEY.asc

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

---

OpenSSH 9.2/9.2p1 (2023-02-02)

OpenSSH 9.2 was released on 2023-02-02. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
https://www.openssh.com/donations.html

Changes since OpenSSH 9.1
=========================

This release fixes a number of security bugs.

Security
========

This release contains fixes for two security problems and a memory
safety problem. The memory safety problem is not believed to be
exploitable, but we report most network-reachable memory faults as
security bugs.

 * sshd(8): fix a pre-authentication double-free memory fault
   introduced in OpenSSH 9.1. This is not believed to be exploitable,
   and it occurs in the unprivileged pre-auth process that is
   subject to chroot(2) and is further sandboxed on most major
   platforms.

 * ssh(8): in OpenSSH releases after 8.7, the PermitRemoteOpen option
   would ignore its first argument unless it was one of the special
   keywords "any" or "none", causing the permission list to fail open
   if only one permission was specified. bz3515

 * ssh(1): if the CanonicalizeHostname and CanonicalizePermittedCNAMEs
   options were enabled, and the system/libc resolver did not check
   that names in DNS responses were valid, then use of these options
   could allow an attacker with control of DNS to include invalid
   characters (possibly including wildcards) in names added to
   known_hosts files when they were updated. These names would still
   have to match the CanonicalizePermittedCNAMEs allow-list, so
   practical exploitation appears unlikely.

Potentially-incompatible changes
----

 * ssh(1): add a new EnableEscapeCommandline ssh_config(5) option that
   controls whether the client-side ~C escape sequence that provides a
   command-line is available. Among other things, the ~C command-line
   could be used to add additional port-forwards at runtime.

   This option defaults to "no", disabling the ~C command-line that
   was previously enabled by default. Turning off the command-line
   allows platforms that support sandboxing of the ssh(1) client
   (currently only OpenBSD) to use a stricter default sandbox policy.

New features
----

 * sshd(8): add support for channel inactivity timeouts via a new
   sshd_config(5) ChannelTimeout directive. This allows channels that
   have not seen traffic in a configurable interval to be
   automatically closed. Different timeouts may be applied to session,
   X11, agent and TCP forwarding channels.

 * sshd(8): add a sshd_config UnusedConnectionTimeout option to
   terminate client connections that have no open channels for a
   length of time. This complements the ChannelTimeout option above.

 * sshd(8): add a -V (version) option to sshd like the ssh client has.

 * ssh(1): add a "Host" line to the output of ssh -G showing the
   original hostname argument. bz3343

 * scp(1), sftp(1): add a -X option to both scp(1) and sftp(1) to
   allow control over some SFTP protocol parameters: the copy buffer
   length and the number of in-flight requests, both of which are used
   during upload/download. Previously these could be controlled in
   sftp(1) only. This makes them available in both SFTP protocol

clients using the same option character sequence.

* ssh-keyscan(1): allow scanning of complete CIDR address ranges, e.g. "ssh-keyscan 192.168.0.0/24". If a CIDR range is passed, then it will be expanded to all possible addresses in the range including the all-0s and all-1s addresses. bz#976

* ssh(1): support dynamic remote port forwarding in escape command-line's -R processing. bz#3499

Bugfixes
----

* ssh(1): when restoring non-blocking mode to stdio fds, restore exactly the flags that ssh started with and don't just clobber them with zero, as this could also remove the append flag from the set. bz3523

* ssh(1): avoid printf("%s", NULL) if using UserKnownHostsFile=none and a hostkey in one of the system known hosts file changes.

* scp(1): switch scp from using pipes to a socket-pair for communication with its ssh sub-processes, matching how sftp(1) operates.

* sshd(8): clear signal mask early in main(); sshd may have been started with one or more signals masked (sigprocmask(2) is not cleared on fork/exec) and this could interfere with various things, e.g. the login grace timer. Execution environments that fail to clear the signal mask before running sshd are clearly broken, but apparently they do exist.

* ssh(1): warn if no host keys for hostbased auth can be loaded.

* sshd(8): Add server debugging for hostbased auth that is queued and sent to the client after successful authentication, but also logged to assist in diagnosis of HostbasedAuthentication problems. bz3507

* ssh(1): document use of the IdentityFile option as being usable to list public keys as well as private keys. GHPR352

* sshd(8): check for and disallow MaxStartups values less than or equal to zero during config parsing, rather than failing later at runtime. bz3489

* ssh-keygen(1): fix parsing of hex cert expiry times specified on the command-line when acting as a CA.

* scp(1): when scp(1) is using the SFTP protocol for transport (the default), better match scp/rcp's handling of globs that don't match the globbed characters but do match literally (e.g. trying to transfer a file named "foo.[1]"). Previously scp(1) in SFTP mode would not match these pathnames but legacy scp/rcp mode would. bz3488

* ssh-agent(1): document the "-O no-restrict-websafe" command-line option.

* ssh(1): honour user's umask(2) if it is more restrictive then the ssh default (022).

Portability
----

* sshd(8): allow writev(2) in the Linux seccomp sandbox. This seems to be used by recent glibcs at least in some configurations during error conditions. bz3512.

* sshd(8): simply handling of SSH_CONNECTION PAM env var, removing global variable and checking the return value from pam_putenv. bz3508

* sshd(8): disable SANDBOX_SECCOMP_FILTER_DEBUG that was mistakenly enabled during the OpenSSH 9.1 release cycle.

* misc: update autotools and regenerate the config files using the latest autotools

* all: use -fzero-call-used-regs=used on clang 15 instead of -fzero-call-used-reg=all, as some versions of clang 15 have miscompile code when it was enabled. bz3475

* sshd(8): defer PRNG seeding until after the initial closefrom(2) call. PRNG seeding will initialize OpenSSL, and some engine providers (e.g. Intel's QAT) will open descriptors for their own use that closefrom(2) could clobber. bz3483

* misc: in the poll(2)/ppoll(2) compatibility code, avoid assuming the layout of fd_set.

* sftp-server(8), ssh-agent(1): fix ptrace(2) disabling on older FreeBSD kernels. Some versions do not support using id 0 to refer to the current PID for procctl, so try again with getpid() explicitly before failing.

* configure.ac: fix -Wstrict-prototypes in configure test code. Clang 16 now warns on this and legacy prototypes will be removed in C23. GHPR355

```
 * configure.ac: fix setres*id checks to work with clang-16. glibc
   has the prototypes for setresuid behind _GNU_SOURCE, and clang 16
   will error out on implicit function definitions. bz3497

Checksums:
==========

- SHA1 (openssh-9.2.tar.gz) = e4b806b7c81b87d6c90afe97b3d016ba6cf3ba1c
- SHA256 (openssh-9.2.tar.gz) = yYe9uaaWSeetXGXOxuaaEiIsLnvITmGW+l5dgMZb9QU=

- SHA1 (openssh-9.2p1.tar.gz) = 3b172b8e971773a7018bbf3231f6589ae539ca4b
- SHA256 (openssh-9.2p1.tar.gz) = P2bb8WVftF9Q4cVtpiqwEhjCKIB7ITONY068351xz0Y=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available from the mirror sites:
https://cdn.openbsd.org/pub/OpenBSD/OpenSSH/RELEASE_KEY.asc

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com
```

OpenSSH 9.1/9.1p1 (2022-10-04)

```
OpenSSH 9.1 was released on 2022-10-04. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
https://www.openssh.com/donations.html

Changes since OpenSSH 9.0
=========================

This release is focused on bug fixing.

Security
========

This release contains fixes for three minor memory safety problems.
None are believed to be exploitable, but we report most memory safety
problems as potential security vulnerabilities out of caution.

 * ssh-keyscan(1): fix a one-byte overflow in SSH- banner processing.
   Reported by Qualys

 * ssh-keygen(1): double free() in error path of file hashing step in
   signing/verify code; GHPR333

 * ssh-keysign(8): double-free in error path introduced in openssh-8.9

Potentially-incompatible changes
--------------------------------

 * The portable OpenSSH project now signs commits and release tags
   using git's recent SSH signature support. The list of developer
   signing keys is included in the repository as .git_allowed_signers
   and is cross-signed using the PGP key that is still used to sign
   release artifacts:
   https://cdn.openbsd.org/pub/OpenBSD/OpenSSH/RELEASE_KEY.asc

 * ssh(1), sshd(8): SetEnv directives in ssh_config and sshd_config
   are now first-match-wins to match other directives. Previously
   if an environment variable was multiply specified the last set
   value would have been used. bz3438

 * ssh-keygen(8): ssh-keygen -A (generate all default host key types)
   will no longer generate DSA keys, as these are insecure and have
   not been used by default for some years.


New features
------------

 * ssh(1), sshd(8): add a RequiredRSASize directive to set a minimum
   RSA key length. Keys below this length will be ignored for user
   authentication and for host authentication in sshd(8).

   ssh(1) will terminate a connection if the server offers an RSA key
   that falls below this limit, as the SSH protocol does not include
   the ability to retry a failed key exchange.

 * sftp-server(8): add a "users-groups-by-id@openssh.com" extension
   request that allows the client to obtain user/group names that
   correspond to a set of uids/gids.

 * sftp(1): use "users-groups-by-id@openssh.com" sftp-server
   extension (when available) to fill in user/group names for
   directory listings.
```

* sftp-server(8): support the "home-directory" extension request defined in draft-ietf-secsh-filexfer-extensions-00. This overlaps a bit with the existing "expand-path@openssh.com", but some other clients support it.

* ssh-keygen(1), sshd(8): allow certificate validity intervals, sshsig verification times and authorized_keys expiry-time options to accept dates in the UTC time zone in addition to the default of interpreting them in the system time zone. YYYYMMDD and YYMMDDHHMM[SS] dates/times will be interpreted as UTC if suffixed with a 'Z' character.

  Also allow certificate validity intervals to be specified in raw seconds-since-epoch as hex value, e.g. -V 0x1234:0x4567890. This is intended for use by regress tests and other tools that call ssh-keygen as part of a CA workflow. bz3468

* sftp(1): allow arguments to the sftp -D option, e.g. sftp -D "/usr/libexec/sftp-server -el debug3"

* ssh-keygen(1): allow the existing -U (use agent) flag to work with "-Y sign" operations, where it will be interpreted to require that the private keys is hosted in an agent; bz3429

Bugfixes
----

* ssh-keygen(1): implement the "verify-required" certificate option. This was already documented when support for user-verified FIDO keys was added, but the ssh-keygen(1) code was missing.

* ssh-agent(1): hook up the restrict_websafe command-line flag; previously the flag was accepted but never actually used.

* sftp(1): improve filename tab completions: never try to complete names to non-existent commands, and better match the completion type (local or remote filename) against the argument position being completed.

* ssh-keygen(1), ssh(1), ssh-agent(1): several fixes to FIDO key handling, especially relating to keys that request user-verification. These should reduce the number of unnecessary PIN prompts for keys that support intrinsic user verification. GHPR302, GHPR329

* ssh-keygen(1): when enrolling a FIDO resident key, check if a credential with matching application and user ID strings already exists and, if so, prompt the user for confirmation before overwriting the credential. GHPR329

* sshd(8): improve logging of errors when opening authorized_keys files. bz2042

* ssh(1): avoid multiplexing operations that could cause SIGPIPE from causing the client to exit early. bz3454

* ssh_config(5), sshd_config(5): clarify that the RekeyLimit directive applies to both transmitted and received data. GHPR328

* ssh-keygen(1): avoid double fclose() in error path.

* sshd(8): log an error if pipe() fails while accepting a connection. bz3447

* ssh(1), ssh-keygen(1): fix possible NULL deref when built without FIDO support. bz3443

* ssh-keyscan(1): add missing *-sk types to ssh-keyscan manpage. GHPR294.

* sshd(8): ensure that authentication passwords are cleared from memory in error paths. GHPR286

* ssh(1), ssh-agent(1): avoid possibility of notifier code executing kill(-1). GHPR286

* ssh_config(5): note that the ProxyJump directive also accepts the same tokens as ProxyCommand. GHPR305.

* scp(1): do not not ftruncate(3) files early when in sftp mode. The previous behaviour of unconditionally truncating the destination file would cause "scp ~/foo localhost:foo" and the reverse "scp localhost:foo ~/foo" to delete all the contents of their destination. bz3431

* ssh-keygen(1): improve error message when 'ssh-keygen -Y sign' is unable to load a private key; bz3429

* sftp(1), scp(1): when performing operations that glob(3) a remote path, ensure that the implicit working directory used to construct that path escapes glob(3) characters. This prevents glob characters from being processed in places they shouldn't, e.g. "cd /tmp/a*/", "get *.txt" should have the get operation treat the path "/tmp/a*" literally and not attempt to expand it.

* ssh(1), sshd(8): be stricter in which characters will be accepted in specifying a mask length; allow only 0-9. GHPR278

```
 * ssh-keygen(1): avoid printing hash algorithm twice when dumping a
   KRL

 * ssh(1), sshd(8): continue running local I/O for open channels
   during SSH transport rekeying. This should make ~-escapes work in
   the client (e.g. to exit) if the connection happened to have
   stalled during a rekey event.

 * ssh(1), sshd(8): avoid potential poll() spin during rekeying

 * Further hardening for sshbuf internals: disallow "reparenting" a
   hierarchical sshbuf and zero the entire buffer if reallocation
   fails. GHPR287

Portability
-----------

 * ssh(1), ssh-keygen(1), sshd(8): automatically enable the built-in
   FIDO security key support if libfido2 is found and usable, unless
   --without-security-key-builtin was requested.

 * ssh(1), ssh-keygen(1), sshd(8): many fixes to make the WinHello
   FIDO device usable on Cygwin. The windows://hello FIDO device will
   be automatically used by default on this platform unless requested
   otherwise, or when probing resident FIDO credentials (an operation
   not currently supported by WinHello).

 * Portable OpenSSH: remove workarounds for obsolete and unsupported
   versions of OpenSSL libcrypto. In particular, this release removes
   fallback support for OpenSSL that lacks AES-CTR or AES-GCM.

   Those AES cipher modes were added to OpenSSL prior to the minimum
   version currently supported by OpenSSH, so this is not expected to
   impact any currently supported configurations.

 * sshd(8): fix SANDBOX_SECCOMP_FILTER_DEBUG on current Linux/glibc

 * All: resync and clean up internal CSPRNG code.

 * scp(1), sftp(1), sftp-server(8): avoid linking these programs with
   unnecessary libraries. They are no longer linked against libz and
   libcrypto. This may be of benefit to space constrained systems
   using any of those components in isolation.

 * sshd(8): add AUDIT_ARCH_PPC to supported seccomp sandbox
   architectures.

 * configure: remove special casing of crypt(). configure will no
   longer search for crypt() in libcrypto, as it was removed from
   there years ago. configure will now only search libc and libcrypt.

 * configure: refuse to use OpenSSL 3.0.4 due to potential RCE in its
   RSA implementation (CVE-2022-2274) on x86_64.

 * All: request 1.1x API compatibility for OpenSSL >=3.x; GHPR#322

 * ssh(1), ssh-keygen(1), sshd(8): fix a number of missing includes
   required by the XMSS code on some platforms.

 * sshd(8): cache timezone data in capsicum sandbox.

Checksums:
==========

- SHA1 (openssh-9.1.tar.gz) = 3ae2d6a3a695d92778c4c4567dcd6ad481092f6c
- SHA256 (openssh-9.1.tar.gz) = QKfVArlcItV+e8V1Th85TL5//5d/AvOUhYOeHMDEGuE=

- SHA1 (openssh-9.1p1.tar.gz) = 15545440268967511d3194ebf20bcd0c7ff3fcc9
- SHA256 (openssh-9.1p1.tar.gz) = GfhQCcfj4jeH8CNvuxV4OSq01L+fjsX+a8HNfov90og=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available from the mirror sites:
https://cdn.openbsd.org/pub/OpenBSD/OpenSSH/RELEASE_KEY.asc

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com
```

OpenSSH 9.0/9.0p1 (2022-04-08)

```
OpenSSH 9.0 was released on 2022-04-08. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
https://www.openssh.com/donations.html

Changes since OpenSSH 8.9
```

```
========================

This release is focused on bug fixing.

Potentially-incompatible changes
--------------------------------

This release switches scp(1) from using the legacy scp/rcp protocol
to using the SFTP protocol by default.

Legacy scp/rcp performs wildcard expansion of remote filenames (e.g.
"scp host:* ."") through the remote shell. This has the side effect of
requiring double quoting of shell meta-characters in file names
included on scp(1) command-lines, otherwise they could be interpreted
as shell commands on the remote side.

This creates one area of potential incompatibility: scp(1) when using
the SFTP protocol no longer requires this finicky and brittle quoting,
and attempts to use it may cause transfers to fail. We consider the
removal of the need for double-quoting shell characters in file names
to be a benefit and do not intend to introduce bug-compatibility for
legacy scp/rcp in scp(1) when using the SFTP protocol.

Another area of potential incompatibility relates to the use of remote
paths relative to other user's home directories, for example -
"scp host:~user/file /tmp". The SFTP protocol has no native way to
expand a ~user path. However, sftp-server(8) in OpenSSH 8.7 and later
support a protocol extension "expand-path@openssh.com" to support
this.

In case of incompatibility, the scp(1) client may be instructed to use
the legacy scp/rcp using the -O flag.

New features
------------

 * ssh(1), sshd(8): use the hybrid Streamlined NTRU Prime + x25519 key
   exchange method by default ("sntrup761x25519-sha512@openssh.com").
   The NTRU algorithm is believed to resist attacks enabled by future
   quantum computers and is paired with the X25519 ECDH key exchange
   (the previous default) as a backstop against any weaknesses in
   NTRU Prime that may be discovered in the future. The combination
   ensures that the hybrid exchange offers at least as good security
   as the status quo.

   We are making this change now (i.e. ahead of cryptographically-
   relevant quantum computers) to prevent "capture now, decrypt
   later" attacks where an adversary who can record and store SSH
   session ciphertext would be able to decrypt it once a sufficiently
   advanced quantum computer is available.

 * sftp-server(8): support the "copy-data" extension to allow server-
   side copying of files/data, following the design in
   draft-ietf-secsh-filexfer-extensions-00. bz2948

 * sftp(1): add a "cp" command to allow the sftp client to perform
   server-side file copies.

Bugfixes
--------

 * ssh(1), sshd(8): upstream: fix poll(2) spin when a channel's output
   fd closes without data in the channel buffer. bz3405 and bz3411

 * sshd(8): pack pollfd array in server listen/accept loop. Could
   cause the server to hang/spin when MaxStartups > RLIMIT_NOFILE

 * ssh-keygen(1): avoid NULL deref via the find-principals and
   check-novalidate operations. bz3409 and GHPR#307 respectively.

 * scp(1): fix a memory leak in argument processing. bz3404

 * sshd(8): don't try to resolve ListenAddress directives in the sshd
   re-exec path. They are unused after re-exec and parsing errors
   (possible for example if the host's network configuration changed)
   could prevent connections from being accepted.

 * sshd(8): when refusing a public key authentication request from a
   client for using an unapproved or unsupported signature algorithm
   include the algorithm name in the log message to make debugging
   easier.

Portability
-----------

 * sshd(8): refactor platform-specific locked account check, fixing
   an incorrect free() on platforms with both libiaf and shadow
   passwords (probably only Unixware) GHPR#284,

 * ssh(1), sshd(8): Fix possible integer underflow in scan_scaled(3)
   parsing of K/M/G/etc quantities. bz#3401.

 * sshd(8): provide killpg implementation (mostly for Tandem NonStop)
   GHPR#301.

 * Check for missing ftruncate prototype. GHPR#301

 * sshd(8): default to not using sandbox when cross compiling. On most
```

```
    systems poll(2) does not work when the number of FDs is reduced
    with setrlimit, so assume it doesn't when cross compiling and we
    can't run the test.  bz#3398.

  * sshd(8): allow ppoll_time64 in seccomp sandbox. Should fix sandbox
    violations on some (at least i386 and armhf) 32bit Linux platforms.
    bz#3396.

  * Improve detection of -fzero-call-used-regs=all support in
    configure script.


Checksums:
==========


 - SHA1 (openssh-9.0.tar.gz) = 05302aa4781e1a69db4261474ed940bd685afc24
 - SHA256 (openssh-9.0.tar.gz) = 9I/FrLf5Gij/4NIPts9A8yWVi0ienyyMqjqn8s0hyLk=

 - SHA1 (openssh-9.0p1.tar.gz) = 06dd658874dcd22d66311cf5999bd56c614de509
 - SHA256 (openssh-9.0p1.tar.gz) = A5dDAhYenszjIVPPoQAS8eZcjzdQ9XOnOrG+/Vlyooo=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available from the mirror sites:
https://cdn.openbsd.org/pub/OpenBSD/OpenSSH/RELEASE_KEY.asc

Please note that the OpenPGP key used to sign releases has been
rotated for this release. The new key has been signed by the previous
key to provide continuity.

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com
```

OpenSSH 8.9/8.9p1 (2022-02-23)

```
OpenSSH 8.9 was released on 2022-02-23. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
https://www.openssh.com/donations.html

Future deprecation notice
=========================

A near-future release of OpenSSH will switch scp(1) from using the
legacy scp/rcp protocol to using SFTP by default.

Legacy scp/rcp performs wildcard expansion of remote filenames (e.g.
"scp host:* .") through the remote shell. This has the side effect of
requiring double quoting of shell meta-characters in file names
included on scp(1) command-lines, otherwise they could be interpreted
as shell commands on the remote side.

This creates one area of potential incompatibility: scp(1) when using
the SFTP protocol no longer requires this finicky and brittle quoting,
and attempts to use it may cause transfers to fail. We consider the
removal of the need for double-quoting shell characters in file names
to be a benefit and do not intend to introduce bug-compatibility for
legacy scp/rcp in scp(1) when using the SFTP protocol.

Another area of potential incompatibility relates to the use of remote
paths relative to other user's home directories, for example -
"scp host:~user/file /tmp". The SFTP protocol has no native way to
expand a ~user path. However, sftp-server(8) in OpenSSH 8.7 and later
support a protocol extension "expand-path@openssh.com" to support
this.

Security Near Miss
==================

 * sshd(8): fix an integer overflow in the user authentication path
   that, in conjunction with other logic errors, could have yielded
   unauthenticated access under difficult to exploit conditions.

   This situation is not exploitable because of independent checks in
   the privilege separation monitor. Privilege separation has been
   enabled by default in since openssh-3.2.2 (released in 2002) and
   has been mandatory since openssh-7.5 (released in 2017). Moreover,
   portable OpenSSH has used toolchain features available in most
   modern compilers to abort on signed integer overflow since
   openssh-6.5 (released in 2014).

   Thanks to Malcolm Stagg for finding and reporting this bug.

Potentially-incompatible changes
================================
```

```
 * sshd(8), portable OpenSSH only: this release removes in-built
   support for MD5-hashed passwords. If you require these on your
   system then we recommend linking against libxcrypt or similar.

 * This release modifies the FIDO security key middleware interface
   and increments SSH_SK_VERSION_MAJOR.

Changes since OpenSSH 8.8
=========================

This release includes a number of new features.

New features
------------

 * ssh(1), sshd(8), ssh-add(1), ssh-agent(1): add a system for
   restricting forwarding and use of keys added to ssh-agent(1)
   A detailed description of the feature is available at
   https://www.openssh.com/agent-restrict.html and the protocol
   extensions are documented in the PROTOCOL and PROTOCOL.agent
   files in the source release.

 * ssh(1), sshd(8): add the sntrup761x25519-sha512@openssh.com hybrid
   ECDH/x25519 + Streamlined NTRU Prime post-quantum KEX to the
   default KEXAlgorithms list (after the ECDH methods but before the
   prime-group DH ones). The next release of OpenSSH is likely to
   make this key exchange the default method.

 * ssh-keygen(1): when downloading resident keys from a FIDO token,
   pass back the user ID that was used when the key was created and
   append it to the filename the key is written to (if it is not the
   default). Avoids keys being clobbered if the user created multiple
   resident keys with the same application string but different user
   IDs.

 * ssh-keygen(1), ssh(1), ssh-agent(1): better handling for FIDO keys
   on tokens that provide user verification (UV) on the device itself,
   including biometric keys, avoiding unnecessary PIN prompts.

 * ssh-keygen(1): add "ssh-keygen -Y match-principals" operation to
   perform matching of principals names against an allowed signers
   file. To be used towards a TOFU model for SSH signatures in git.

 * ssh-add(1), ssh-agent(1): allow pin-required FIDO keys to be added
   to ssh-agent(1). $SSH_ASKPASS will be used to request the PIN at
   authentication time.

 * ssh-keygen(1): allow selection of hash at sshsig signing time
   (either sha512 (default) or sha256).

 * ssh(1), sshd(8): read network data directly to the packet input
   buffer instead of indirectly via a small stack buffer. Provides a
   modest performance improvement.

 * ssh(1), sshd(8): read data directly to the channel input buffer,
   providing a similar modest performance improvement.

 * ssh(1): extend the PubkeyAuthentication configuration directive to
   accept yes|no|unbound|host-bound to allow control over one of the
   protocol extensions used to implement agent-restricted keys.

Bugfixes
--------

 * sshd(8): document that CASignatureAlgorithms, ExposeAuthInfo and
   PubkeyAuthOptions can be used in a Match block. PR#277.

 * sshd(8): fix possible string truncation when constructing paths to
   .rhosts/.shosts files with very long user home directory names.

 * ssh-keysign(1): unbreak for KEX algorithms that use SHA384/512
   exchange hashes

 * ssh(1): don't put the TTY into raw mode when SessionType=none,
   avoids ^C being unable to kill such a session. bz3360

 * scp(1): fix some corner-case bugs in SFTP-mode handling of
   ~-prefixed paths.

 * ssh(1): unbreak hostbased auth using RSA keys. Allow ssh(1) to
   select RSA keys when only RSA/SHA2 signature algorithms are
   configured (this is the default case). Previously RSA keys were
   not being considered in the default case.

 * ssh-keysign(1): make ssh-keysign use the requested signature
   algorithm and not the default for the key type. Part of unbreaking
   hostbased auth for RSA/SHA2 keys.

 * ssh(1): stricter UpdateHostkey signature verification logic on
   the client- side. Require RSA/SHA2 signatures for RSA hostkeys
   except when RSA/SHA1 was explicitly negotiated during initial
   KEX; bz3375

 * ssh(1), sshd(8): fix signature algorithm selection logic for
   UpdateHostkeys on the server side. The previous code tried to
   prefer RSA/SHA2 for hostkey proofs of RSA keys, but missed some
   cases. This will use RSA/SHA2 signatures for RSA keys if the
```

client proposed these algorithms in initial KEX. bz3375

* All: convert all uses of select(2)/pselect(2) to poll(2)/ppoll(2). This includes the mainloops in ssh(1), ssh-agent(1), ssh-agent(1) and sftp-server(8), as well as the sshd(8) listen loop and all other FD read/writability checks. On platforms with missing or broken poll(2)/ppoll(2) syscalls a select(2)-based compat shim is available.

* ssh-keygen(1): the "-Y find-principals" command was verifying key validity when using ca certs but not with simple key lifetimes within the allowed signers file.

* ssh-keygen(1): make sshsig verify-time argument parsing optional

* sshd(8): fix truncation in rhosts/shosts path construction.

* ssh(1), ssh-agent(1): avoid xmalloc(0) for PKCS#11 keyid for ECDSA keys (we already did this for RSA keys). Avoids fatal errors for PKCS#11 libraries that return empty keyid, e.g. Microchip ATECC608B "cryptoauthlib"; bz#3364

* ssh(1), ssh-agent(1): improve the testing of credentials against inserted FIDO: ask the token whether a particular key belongs to it in cases where the token supports on-token user-verification (e.g. biometrics) rather than just assuming that it will accept it.

  Will reduce spurious "Confirm user presence" notifications for key handles that relate to FIDO keys that are not currently inserted in at least some cases. bz3366

* ssh(1), sshd(8): correct value for IPTOS_DSCP_LE. It needs to allow for the preceding two ECN bits. bz#3373

* ssh-keygen(1): add missing -O option to usage() for the "-Y sign" option.

* ssh-keygen(1): fix a NULL deref when using the find-principals function, when matching an allowed_signers line that contains a namespace restriction, but no restriction specified on the command-line

* ssh-agent(1): fix memleak in process_extension(); oss-fuzz issue #42719

* ssh(1): suppress "Connection to xxx closed" messages when LogLevel is set to "error" or above. bz3378

* ssh(1), sshd(8): use correct zlib flags when inflate(3)-ing compressed packet data. bz3372

* scp(1): when recursively transferring files in SFTP mode, create the destination directory if it doesn't already exist to match scp(1) in legacy RCP mode behaviour.

* scp(1): many improvements in error message consistency between scp(1) in SFTP mode vs legacy RCP mode.

* sshd(8): fix potential race in SIGTERM handling PR#289

* ssh(1), ssh(8): since DSA keys are deprecated, move them to the end of the default list of public keys so that they will be tried last. PR#295

* ssh-keygen(1): allow 'ssh-keygen -Y find-principals' to match wildcard principals in allowed_signers files

Portability
---

* ssh(1), sshd(8): don't trust closefrom(2) on Linux. glibc's implementation does not work in a chroot when the kernel does not have close_range(2). It tries to read from /proc/self/fd and when that fails dies with an assertion of sorts. Instead, call close_range(2) directly from our compat code and fall back if that fails. bz#3349,

* OS X poll(2) is broken; use compat replacement. For character-special devices like /dev/null, Darwin's poll(2) returns POLLNVAL when polled with POLLIN. Apparently this is Apple bug 3710161 - not public but a websearch will find other OSS projects rediscovering it periodically since it was first identified in 2005.

* Correct handling of exceptfds/POLLPRI in our select(2)-based poll(2)/ppoll(2) compat implementation.

* Cygwin: correct checking of mbstowcs() return value.

* Add a basic SECURITY.md that refers people to the openssh.com website.

* Enable additional compiler warnings and toolchain hardening flags, including -Wbitwise-instead-of-logical, -Wmisleading-indentation, -fzero-call-used-regs and -ftrivial-auto-var-init.

* HP/UX. Use compat getline(3) on HP-UX 10.x, where the libc version is not reliable.

```
Checksums:
==========

 - SHA1 (openssh-8.9.tar.gz) = 653310ba1a63959fe2df503fe7ad556445180127
 - SHA256 (openssh-8.9.tar.gz) = mJigktP+Bk0sB7uRPuWgjcCOYZ+mIMdvRlZe66irtQA=

 - SHA1 (openssh-8.9p1.tar.gz) = 205cdf0040a238047e2c49f43460e03d76e5d650
 - SHA256 (openssh-8.9p1.tar.gz) = /Ul2VLerFobaxnL7g9+0ukCW6LX/zazNJiOArli+xec=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available from the mirror sites:
https://cdn.openbsd.org/pub/OpenBSD/OpenSSH/RELEASE_KEY.asc

Please note that the OpenPGP key used to sign releases has been
rotated for this release. The new key has been signed by the previous
key to provide continuity.

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com
```

OpenSSH 8.8/8.8p1 (2021-09-26)

```
OpenSSH 8.8 was released on 2021-09-26. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
https://www.openssh.com/donations.html

Future deprecation notice
=========================

A near-future release of OpenSSH will switch scp(1) from using the
legacy scp/rcp protocol to using SFTP by default.

Legacy scp/rcp performs wildcard expansion of remote filenames (e.g.
"scp host:* .") through the remote shell. This has the side effect of
requiring double quoting of shell meta-characters in file names
included on scp(1) command-lines, otherwise they could be interpreted
as shell commands on the remote side.

This creates one area of potential incompatibility: scp(1) when using
the SFTP protocol no longer requires this finicky and brittle quoting,
and attempts to use it may cause transfers to fail. We consider the
removal of the need for double-quoting shell characters in file names
to be a benefit and do not intend to introduce bug- compatibility for
legacy scp/rcp in scp(1) when using the SFTP protocol.

Another area of potential incompatibility relates to the use of remote
paths relative to other user's home directories, for example -
"scp host:~user/file /tmp". The SFTP protocol has no native way to
expand a ~user path. However, sftp-server(8) in OpenSSH 8.7 and later
support a protocol extension "expand-path@openssh.com" to support
this.

Security
========

sshd(8) from OpenSSH 6.2 through 8.7 failed to correctly initialise
supplemental groups when executing an AuthorizedKeysCommand or
AuthorizedPrincipalsCommand, where a AuthorizedKeysCommandUser or
AuthorizedPrincipalsCommandUser directive has been set to run the
command as a different user. Instead these commands would inherit
the groups that sshd(8) was started with.

Depending on system configuration, inherited groups may allow
AuthorizedKeysCommand/AuthorizedPrincipalsCommand helper programs to
gain unintended privilege.

Neither AuthorizedKeysCommand nor AuthorizedPrincipalsCommand are
enabled by default in sshd_config(5).

Potentially-incompatible changes
================================

This release disables RSA signatures using the SHA-1 hash algorithm
by default. This change has been made as the SHA-1 hash algorithm is
cryptographically broken, and it is possible to create chosen-prefix
hash collisions for <USD$50K [1]

For most users, this change should be invisible and there is
no need to replace ssh-rsa keys. OpenSSH has supported RFC8332
RSA/SHA-256/512 signatures since release 7.2 and existing ssh-rsa keys
will automatically use the stronger algorithm where possible.

Incompatibility is more likely when connecting to older SSH
implementations that have not been upgraded or have not closely tracked
```

improvements in the SSH protocol. For these cases, it may be necessary
to selectively re-enable RSA/SHA1 to allow connection and/or user
authentication via the HostkeyAlgorithms and PubkeyAcceptedAlgorithms
options. For example, the following stanza in ~/.ssh/config will enable
RSA/SHA1 for host and user authentication for a single destination host:

```
    Host old-host
        HostkeyAlgorithms +ssh-rsa
        PubkeyAcceptedAlgorithms +ssh-rsa
```

We recommend enabling RSA/SHA1 only as a stopgap measure until legacy
implementations can be upgraded or reconfigured with another key type
(such as ECDSA or Ed25519).

[1] "SHA-1 is a Shambles: First Chosen-Prefix Collision on SHA-1 and
    Application to the PGP Web of Trust" Leurent, G and Peyrin, T
    (2020) https://eprint.iacr.org/2020/014.pdf

Changes since OpenSSH 8.7
=========================

This release is motivated primarily by the above deprecation and
security fix.

New features
----

 * ssh(1): allow the ssh_config(5) CanonicalizePermittedCNAMEs
   directive to accept a "none" argument to specify the default
   behaviour.

Bugfixes
----

 * scp(1): when using the SFTP protocol, continue transferring files
   after a transfer error occurs, better matching original scp/rcp
   behaviour.

 * ssh(1): fixed a number of memory leaks in multiplexing,

 * ssh-keygen(1): avoid crash when using the -Y find-principals
   command.

 * A number of documentation and manual improvements, including
   bz#3340, PR#139, PR#215, PR#241, PR#257

Portability
----

 * ssh-agent(1): on FreeBSD, use procctl to disable ptrace(2)

 * ssh(1)/sshd(8): some fixes to the pselect(2) replacement
   compatibility code. bz#3345

Checksums:
==========

 - SHA1 (openssh-8.8.tar.gz) = 732947082a8998047e839cc0b4c066bf0a7e1a5b
 - SHA256 (openssh-8.8.tar.gz) = AngyrPSQH255hnzU1l7y+LlVAUNcGWtuYQIFEl22nRo=

 - SHA1 (openssh-8.8p1.tar.gz) = 1eb964897a4372f6fb96c7effeb509ec71c379c9
 - SHA256 (openssh-8.8p1.tar.gz) = RZCJDqm7ms5Pca4zF4WjpYIyMkNRYZYO1fyGWI8zH+k=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available from the mirror sites:
https://cdn.openbsd.org/pub/OpenBSD/OpenSSH/RELEASE_KEY.asc

Please note that the OpenPGP key used to sign releases has been
rotated for this release. The new key has been signed by the previous
key to provide continuity.

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

---

OpenSSH 8.7/8.7p1 (2021-08-20)

OpenSSH 8.7 was released on 2021-08-20. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
https://www.openssh.com/donations.html

Imminent deprecation notice
===========================

OpenSSH will disable the ssh-rsa signature scheme by default in the
next release.

In the SSH protocol, the "ssh-rsa" signature scheme uses the SHA-1
hash algorithm in conjunction with the RSA public key algorithm.
It is now possible[1] to perform chosen-prefix attacks against the
SHA-1 algorithm for less than USD$50K.

Note that the deactivation of "ssh-rsa" signatures does not necessarily
require cessation of use for RSA keys. In the SSH protocol, keys may be
capable of signing using multiple algorithms. In particular, "ssh-rsa"
keys are capable of signing using "rsa-sha2-256" (RSA/SHA256),
"rsa-sha2-512" (RSA/SHA512) and "ssh-rsa" (RSA/SHA1). Only the last of
these is being turned off by default.

This algorithm is unfortunately still used widely despite the
existence of better alternatives, being the only remaining public key
signature algorithm specified by the original SSH RFCs that is still
enabled by default.

The better alternatives include:

 * The RFC8332 RSA SHA-2 signature algorithms rsa-sha2-256/512. These
   algorithms have the advantage of using the same key type as
   "ssh-rsa" but use the safe SHA-2 hash algorithms. These have been
   supported since OpenSSH 7.2 and are already used by default if the
   client and server support them.

 * The RFC8709 ssh-ed25519 signature algorithm. It has been supported
   in OpenSSH since release 6.5.

 * The RFC5656 ECDSA algorithms: ecdsa-sha2-nistp256/384/521. These
   have been supported by OpenSSH since release 5.7.

To check whether a server is using the weak ssh-rsa public key
algorithm, for host authentication, try to connect to it after
removing the ssh-rsa algorithm from ssh(1)'s allowed list:

    ssh -oHostKeyAlgorithms=-ssh-rsa user@host

If the host key verification fails and no other supported host key
types are available, the server software on that host should be
upgraded.

OpenSSH recently enabled the UpdateHostKeys option by default to
assist the client by automatically migrating to better algorithms.

[1] "SHA-1 is a Shambles: First Chosen-Prefix Collision on SHA-1 and
    Application to the PGP Web of Trust" Leurent, G and Peyrin, T
    (2020) https://eprint.iacr.org/2020/014.pdf

Potentially-incompatible changes
================================

This release includes a number of changes that may affect existing
configurations:

 * scp(1): this release changes the behaviour of remote to remote
   copies (e.g. "scp host-a:/path host-b:") to transfer through the
   local host by default. This was previously available via the -3
   flag. This mode avoids the need to expose credentials on the
   origin hop, avoids triplicate interpretation of filenames by the
   shell (by the local system, the copy origin and the destination)
   and, in conjunction with the SFTP support for scp(1) mentioned
   below, allows use of all authentication methods to the remote
   hosts (previously, only non-interactive methods could be used).
   A -R flag has been added to select the old behaviour.

 * ssh(1)/sshd(8): both the client and server are now using a
   stricter configuration file parser. The new parser uses more
   shell-like rules for quotes, space and escape characters. It is
   also more strict in rejecting configurations that include options
   lacking arguments. Previously some options (e.g. DenyUsers) could
   appear on a line with no subsequent arguments. This release will
   reject such configurations. The new parser will also reject
   configurations with unterminated quotes and multiple '='
   characters after the option name.

 * ssh(1): when using SSHFP DNS records for host key verification,
   ssh(1) will verify all matching records instead of just those
   with the specific signature type requested. This may cause host
   key verification problems if stale SSHFP records of a different
   or legacy signature type exist alongside other records for a
   particular host. bz#3322

 * ssh-keygen(1): when generating a FIDO key and specifying an
   explicit attestation challenge (using -Ochallenge), the challenge
   will now be hashed by the builtin security key middleware. This
   removes the (undocumented) requirement that challenges be exactly
   32 bytes in length and matches the expectations of libfido2.

 * sshd(8): environment="..." directives in authorized_keys files are
   now first-match-wins and limited to 1024 discrete environment
   variable names.

Changes since OpenSSH 8.6
=========================

This release contains a mix of new features and bug-fixes.

New features

- scp(1): experimental support for transfers using the SFTP protocol as a replacement for the venerable SCP/RCP protocol that it has traditionally used. SFTP offers more predictable filename handling and does not require expansion of glob(3) patterns via the shell on the remote side.

  SFTP support may be enabled via a temporary scp -s flag. It is intended for SFTP to become the default transfer mode in the near future, at which time the -s flag will be removed. The -O flag exists to force use of the original SCP/RCP protocol for cases where SFTP may be unavailable or incompatible.

- sftp-server(8): add a protocol extension to support expansion of ~/ and ~user/ prefixed paths. This was added to support these paths when used by scp(1) while in SFTP mode.

- ssh(1): add a ForkAfterAuthentication ssh_config(5) counterpart to the ssh(1) -f flag. GHPR#231

- ssh(1): add a StdinNull directive to ssh_config(5) that allows the config file to do the same thing as -n does on the ssh(1) command-line. GHPR#231

- ssh(1): add a SessionType directive to ssh_config, allowing the configuration file to offer equivalent control to the -N (no session) and -s (subsystem) command-line flags. GHPR#231

- ssh-keygen(1): allowed signers files used by ssh-keygen(1) signatures now support listing key validity intervals alongside they key, and ssh-keygen(1) can optionally check during signature verification whether a specified time falls inside this interval. This feature is intended for use by git to support signing and verifying objects using ssh keys.

- ssh-keygen(8): support printing of the full public key in a sshsig signature via a -Oprint-pubkey flag.

Bugfixes
----

* ssh(1)/sshd(8): start time-based re-keying exactly on schedule in the client and server mainloops. Previously the re-key timeout could expire but re-keying would not start until a packet was sent or received, causing a spin in select() if the connection was quiescent.

* ssh-keygen(1): avoid Y2038 problem in printing certificate validity lifetimes. Dates past 2^31-1 seconds since epoch were displayed incorrectly on some platforms. bz#3329

* scp(1): allow spaces to appear in usernames for local to remote and scp -3 remote to remote copies. bz#1164

* ssh(1)/sshd(8): remove references to ChallengeResponseAuthentication in favour of KbdInteractiveAuthentication. The former is what was in SSHv1, the latter is what is in SSHv2 (RFC4256) and they were treated as somewhat but not entirely equivalent. We retain the old name as a deprecated alias so configuration files continue to work as well as a reference in the man page for people looking for it. bz#3303

* ssh(1)/ssh-add(1)/ssh-keygen(1): fix decoding of X.509 subject name when extracting a key from a PKCS#11 certificate. bz#3327

* ssh(1): restore blocking status on stdio fds before close. ssh(1) needs file descriptors in non-blocking mode to operate but it was not restoring the original state on exit. This could cause problems with fds shared with other programs via the shell, bz#3280 and GHPR#246

* ssh(1)/sshd(8): switch both client and server mainloops from select(3) to pselect(3). Avoids race conditions where a signal may arrive immediately before select(3) and not be processed until an event fires. bz#2158

* ssh(1): sessions started with ControlPersist were incorrectly executing a shell when the -N (no shell) option was specified. bz#3290

* ssh(1): check if IPQoS or TunnelDevice are already set before overriding. Prevents values in config files from overriding values supplied on the command line. bz#3319

* ssh(1): fix debug message when finding a private key to match a certificate being attempted for user authentication. Previously it would print the certificate's path, whereas it was supposed to be showing the private key's path. GHPR#247

* sshd(8): match host certificates against host public keys, not private keys. Allows use of certificates with private keys held in a ssh-agent. bz#3524

* ssh(1): add a workaround for a bug in OpenSSH 7.4 sshd(8), which allows RSA/SHA2 signatures for public key authentication but fails to advertise this correctly via SSH2_MSG_EXT_INFO. This causes

clients of these server to incorrectly match
PubkeyAcceptedAlgorithmse and potentially refuse to offer valid
keys. bz#3213

* sftp(1)/scp(1): degrade gracefully if a sftp-server offers the
limits@openssh.com extension but fails when the client tries to
invoke it. bz#3318

* ssh(1): allow ssh_config SetEnv to override $TERM, which is
otherwise handled specially by the protocol. Useful in ~/.ssh/config
to set TERM to something generic (e.g. "xterm" instead of
"xterm-256color") for destinations that lack terminfo entries.

* sftp-server(8): the limits@openssh.com extension was incorrectly
marked as an operation that writes to the filesystem, which made it
unavailable in sftp-server read-only mode. bz#3318

* ssh(1): fix SEGV in UpdateHostkeys debug() message, triggered when
the update removed more host keys than remain present.

* many manual page fixes.

Portability
----

* ssh(1): move closefrom() to before first malloc. When built against
tcmalloc, the closefrom() would stomp on file descriptors created
for tcmalloc's internal use. bz#3321

* sshd(8): handle GIDs > 2^31 in getgrouplist. When compiled in 32bit
mode, the getgrouplist implementation may fail for GIDs greater than
LONG_MAX.

* ssh(1): xstrdup environment variable used by ForwardAgent. bz#3328

* sshd(8): don't sigdie() in signal handler in privsep child process;
this can end up causing sandbox violations per bz3286

Checksums:
==========

- SHA1 (openssh-8.7.tar.gz) = 61bfa5a55e2b7e851b1d463aa432e4ff508f61cc
- SHA256 (openssh-8.7.tar.gz) = Q5jPfCaYhTKhkDPYH32jtjFN7qhNhkNHYC9awg2THLs=

- SHA1 (openssh-8.7p1.tar.gz) = 8719032c1e47732c8fdb14adfb24b5e9e71de802
- SHA256 (openssh-8.7p1.tar.gz) = fKNLi7JK6eUPM3krcJGzhB1+G0QP9XvJ+r3fAeLtHiQ=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available from the mirror sites:
https://cdn.openbsd.org/pub/OpenBSD/OpenSSH/RELEASE_KEY.asc

Please note that the OpenPGP key used to sign releases has been
rotated for this release. The new key has been signed by the previous
key to provide continuity.

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
Security bugs should be reported directly to openssh@openssh.com

---

OpenSSH 8.6/8.6p1 (2021-04-19)

OpenSSH 8.6 was released on 2021-04-19. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
https://www.openssh.com/donations.html

Future deprecation notice
=========================

It is now possible[1] to perform chosen-prefix attacks against the
SHA-1 algorithm for less than USD$50K.

In the SSH protocol, the "ssh-rsa" signature scheme uses the SHA-1
hash algorithm in conjunction with the RSA public key algorithm.
OpenSSH will disable this signature scheme by default in the near
future.

Note that the deactivation of "ssh-rsa" signatures does not necessarily
require cessation of use for RSA keys. In the SSH protocol, keys may be
capable of signing using multiple algorithms. In particular, "ssh-rsa"
keys are capable of signing using "rsa-sha2-256" (RSA/SHA256),
"rsa-sha2-512" (RSA/SHA512) and "ssh-rsa" (RSA/SHA1). Only the last of
these is being turned off by default.

This algorithm is unfortunately still used widely despite the

existence of better alternatives, being the only remaining public key signature algorithm specified by the original SSH RFCs that is still enabled by default.

The better alternatives include:

 * The RFC8332 RSA SHA-2 signature algorithms rsa-sha2-256/512. These algorithms have the advantage of using the same key type as "ssh-rsa" but use the safe SHA-2 hash algorithms. These have been supported since OpenSSH 7.2 and are already used by default if the client and server support them.

 * The RFC8709 ssh-ed25519 signature algorithm. It has been supported in OpenSSH since release 6.5.

 * The RFC5656 ECDSA algorithms: ecdsa-sha2-nistp256/384/521. These have been supported by OpenSSH since release 5.7.

To check whether a server is using the weak ssh-rsa public key algorithm, for host authentication, try to connect to it after removing the ssh-rsa algorithm from ssh(1)'s allowed list:

    ssh -oHostKeyAlgorithms=-ssh-rsa user@host

If the host key verification fails and no other supported host key types are available, the server software on that host should be upgraded.

OpenSSH recently enabled the UpdateHostKeys option by default to assist the client by automatically migrating to better algorithms.

[1] "SHA-1 is a Shambles: First Chosen-Prefix Collision on SHA-1 and Application to the PGP Web of Trust" Leurent, G and Peyrin, T (2020) https://eprint.iacr.org/2020/014.pdf

Security
========

 * sshd(8): OpenSSH 8.5 introduced the LogVerbose keyword. When this option was enabled with a set of patterns that activated logging in code that runs in the low-privilege sandboxed sshd process, the log messages were constructed in such a way that printf(3) format strings could effectively be specified the low-privilege code.

   An attacker who had sucessfully exploited the low-privilege process could use this to escape OpenSSH's sandboxing and attack the high-privilege process. Exploitation of this weakness is highly unlikely in practice as the LogVerbose option is not enabled by default and is typically only used for debugging. No vulnerabilities in the low-privilege process are currently known to exist.

   Thanks to Ilja Van Sprundel for reporting this bug.

Changes since OpenSSH 8.5
=========================

This release contains mostly bug fixes.

New features
----

 * sftp-server(8): add a new limits@openssh.com protocol extension that allows a client to discover various server limits, including maximum packet size and maximum read/write length.

 * sftp(1): use the new limits@openssh.com extension (when available) to select better transfer lengths in the client.

 * sshd(8): Add ModuliFile keyword to sshd_config to specify the location of the "moduli" file containing the groups for DH-GEX.

 * unit tests: Add a TEST_SSH_ELAPSED_TIMES environment variable to enable printing of the elapsed time in seconds of each test.

Bugfixes
----

 * ssh_config(5), sshd_config(5): sync CASignatureAlgorithms lists in manual pages with the current default. GHPR#174

 * ssh(1): ensure that pkcs11_del_provider() is called before exit. GHPR#234

 * ssh(1), sshd(8): fix problems in string->argv conversion. Multiple backslashes were not being dequoted correctly and quoted space in the middle of a string was being incorrectly split. GHPR#223

 * ssh(1): return non-zero exit status when killed by signal; bz#3281

 * sftp-server(8): increase maximum SSH2_FXP_READ to match the maximum packet size. Also handle zero-length reads that are not explicitly banned by the spec.

Portability
----

 * sshd(8): don't mistakenly exit on transient read errors on the

```
    network socket (e.g. EINTR, EAGAIN); bz3297

  * Create a dedicated contrib/gnome-ssk-askpass3.c source instead of
    building it from the same file as used for GNOME2. Use the GNOME3
    gdk_seat_grab() to manage keyboard/mouse/server grabs for better
    compatibility with Wayland.

  * Fix portability build errors bz3293 bz3292 bz3291 bz3278

  * sshd(8): soft-disallow the fstatat64 syscall in the Linux
    seccomp-bpf sandbox. bz3276

  * unit tests: enable autoopt and misc unit tests that were
    previously skipped

Checksums:
==========

 - SHA1 (openssh-8.6.tar.gz) = a3e93347eed6296faaaceb221e8786391530fccb
 - SHA256 (openssh-8.6.tar.gz) = ihmgdEgKfCBRpC0qzdQRwYownrpBf+rsihvk4Rmim8M=

 - SHA1 (openssh-8.6p1.tar.gz) = 8f9f0c94317baeb97747d6258f3997b4542762c0
 - SHA256 (openssh-8.6p1.tar.gz) = w+bk2hYhdiyFDQO0fu0eSN/0zJYI3etUcgKiNN+O164=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available from the mirror sites:
https://cdn.openbsd.org/pub/OpenBSD/OpenSSH/RELEASE_KEY.asc

Please note that the OpenPGP key used to sign releases has been
rotated for this release. The new key has been signed by the previous
key to provide continuity.

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com
```

OpenSSH 8.5/8.5p1 (2021-03-03)

```
OpenSSH 8.5 was released on 2021-03-03. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
https://www.openssh.com/donations.html

Future deprecation notice
=========================

It is now possible[1] to perform chosen-prefix attacks against the
SHA-1 algorithm for less than USD$50K.

In the SSH protocol, the "ssh-rsa" signature scheme uses the SHA-1
hash algorithm in conjunction with the RSA public key algorithm.
OpenSSH will disable this signature scheme by default in the near
future.

Note that the deactivation of "ssh-rsa" signatures does not necessarily
require cessation of use for RSA keys. In the SSH protocol, keys may be
capable of signing using multiple algorithms. In particular, "ssh-rsa"
keys are capable of signing using "rsa-sha2-256" (RSA/SHA256),
"rsa-sha2-512" (RSA/SHA512) and "ssh-rsa" (RSA/SHA1). Only the last of
these is being turned off by default.

This algorithm is unfortunately still used widely despite the
existence of better alternatives, being the only remaining public key
signature algorithm specified by the original SSH RFCs that is still
enabled by default.

The better alternatives include:

 * The RFC8332 RSA SHA-2 signature algorithms rsa-sha2-256/512. These
   algorithms have the advantage of using the same key type as
   "ssh-rsa" but use the safe SHA-2 hash algorithms. These have been
   supported since OpenSSH 7.2 and are already used by default if the
   client and server support them.

 * The RFC8709 ssh-ed25519 signature algorithm. It has been supported
   in OpenSSH since release 6.5.

 * The RFC5656 ECDSA algorithms: ecdsa-sha2-nistp256/384/521. These
   have been supported by OpenSSH since release 5.7.

To check whether a server is using the weak ssh-rsa public key
algorithm, for host authentication, try to connect to it after
removing the ssh-rsa algorithm from ssh(1)'s allowed list:

    ssh -oHostKeyAlgorithms=-ssh-rsa user@host

If the host key verification fails and no other supported host key
```

types are available, the server software on that host should be upgraded.

This release enables the UpdateHostKeys option by default to assist the client by automatically migrating to better algorithms.

[1] "SHA-1 is a Shambles: First Chosen-Prefix Collision on SHA-1 and Application to the PGP Web of Trust" Leurent, G and Peyrin, T (2020) https://eprint.iacr.org/2020/014.pdf

Security
========

* ssh-agent(1): fixed a double-free memory corruption that was introduced in OpenSSH 8.2 . We treat all such memory faults as potentially exploitable. This bug could be reached by an attacker with access to the agent socket.

  On modern operating systems where the OS can provide information about the user identity connected to a socket, OpenSSH ssh-agent and sshd limit agent socket access only to the originating user and root. Additional mitigation may be afforded by the system's malloc(3)/free(3) implementation, if it detects double-free conditions.

  The most likely scenario for exploitation is a user forwarding an agent either to an account shared with a malicious user or to a host with an attacker holding root access.

* Portable sshd(8): Prevent excessively long username going to PAM. This is a mitigation for a buffer overflow in Solaris' PAM username handling (CVE-2020-14871), and is only enabled for Sun-derived PAM implementations. This is not a problem in sshd itself, it only prevents sshd from being used as a vector to attack Solaris' PAM. It does not prevent the bug in PAM from being exploited via some other PAM application. GHPR#212

Potentially-incompatible changes
================================

This release includes a number of changes that may affect existing configurations:

* ssh(1), sshd(8): this release changes the first-preference signature algorithm from ECDSA to ED25519.

* ssh(1), sshd(8): set the TOS/DSCP specified in the configuration for interactive use prior to TCP connect. The connection phase of the SSH session is time-sensitive and often explicitly interactive. The ultimate interactive/bulk TOS/DSCP will be set after authentication completes.

* ssh(1), sshd(8): remove the pre-standardization cipher rijndael-cbc@lysator.liu.se. It is an alias for aes256-cbc before it was standardized in RFC4253 (2006), has been deprecated and disabled by default since OpenSSH 7.2 (2016) and was only briefly documented in ssh.1 in 2001.

* ssh(1), sshd(8): update/replace the experimental post-quantum hybrid key exchange method based on Streamlined NTRU Prime coupled with X25519.

  The previous sntrup4591761x25519-sha512@tinyssh.org method is replaced with sntrup761x25519-sha512@openssh.com. Per its designers, the sntrup4591761 algorithm was superseded almost two years ago by sntrup761.

  (note this both the updated method and the one that it replaced are disabled by default)

* ssh(1): disable CheckHostIP by default. It provides insignificant benefits while making key rotation significantly more difficult, especially for hosts behind IP-based load-balancers.

Changes since OpenSSH 8.4
=========================

New features
----

* ssh(1): this release enables UpdateHostkeys by default subject to some conservative preconditions:
  - The key was matched in the UserKnownHostsFile (and not in the GlobalKnownHostsFile).
  - The same key does not exist under another name.
  - A certificate host key is not in use.
  - known_hosts contains no matching wildcard hostname pattern.
  - VerifyHostKeyDNS is not enabled.
  - The default UserKnownHostsFile is in use.

  We expect some of these conditions will be modified or relaxed in future.

* ssh(1), sshd(8): add a new LogVerbose configuration directive for that allows forcing maximum debug logging by file/function/line pattern-lists.

* ssh(1): when prompting the user to accept a new hostkey, display any other host names/addresses already associated with the key.

* ssh(1): allow UserKnownHostsFile=none to indicate that no known_hosts file should be used to identify host keys.

* ssh(1): add a ssh_config KnownHostsCommand option that allows the client to obtain known_hosts data from a command in addition to the usual files.

* ssh(1): add a ssh_config PermitRemoteOpen option that allows the client to restrict the destination when RemoteForward is used with SOCKS.

* ssh(1): for FIDO keys, if a signature operation fails with a "incorrect PIN" reason and no PIN was initially requested from the user, then request a PIN and retry the operation. This supports some biometric devices that fall back to requiring PIN when reading of the biometric failed, and devices that require PINs for all hosted credentials.

* sshd(8): implement client address-based rate-limiting via new sshd_config(5) PerSourceMaxStartups and PerSourceNetBlockSize directives that provide more fine-grained control on a per-origin address basis than the global MaxStartups limit.

Bugfixes
----

* ssh(1): Prefix keyboard interactive prompts with "(user@host)" to make it easier to determine which connection they are associated with in cases like scp -3, ProxyJump, etc. bz#3224

* sshd(8): fix sshd_config SetEnv directives located inside Match blocks. GHPR#201

* ssh(1): when requesting a FIDO token touch on stderr, inform the user once the touch has been recorded.

* ssh(1): prevent integer overflow when ridiculously large ConnectTimeout values are specified, capping the effective value (for most platforms) at 24 days. bz#3229

* ssh(1): consider the ECDSA key subtype when ordering host key algorithms in the client.

* ssh(1), sshd(8): rename the PubkeyAcceptedKeyTypes keyword to PubkeyAcceptedAlgorithms. The previous name incorrectly suggested that it control allowed key algorithms, when this option actually specifies the signature algorithms that are accepted. The previous name remains available as an alias. bz#3253

* ssh(1), sshd(8): similarly, rename HostbasedKeyTypes (ssh) and HostbasedAcceptedKeyTypes (sshd) to HostbasedAcceptedAlgorithms.

* sftp-server(8): add missing lsetstat@openssh.com documentation and advertisement in the server's SSH2_FXP_VERSION hello packet.

* ssh(1), sshd(8): more strictly enforce KEX state-machine by banning packet types once they are received. Fixes memleak caused by duplicate SSH2_MSG_KEX_DH_GEX_REQUEST (oss-fuzz #30078).

* sftp(1): allow the full range of UIDs/GIDs for chown/chgrp on 32bit platforms instead of being limited by LONG_MAX. bz#3206

* Minor man page fixes (capitalization, commas, etc.) bz#3223

* sftp(1): when doing an sftp recursive upload or download of a read-only directory, ensure that the directory is created with write and execute permissions in the interim so that the transfer can actually complete, then set the directory permission as the final step. bz#3222

* ssh-keygen(1): document the -Z, check the validity of its argument earlier and provide a better error message if it's not correct. bz#2879

* ssh(1): ignore comments at the end of config lines in ssh_config, similar to what we already do for sshd_config. bz#2320

* sshd_config(5): mention that DisableForwarding is valid in a sshd_config Match block. bz3239

* sftp(1): fix incorrect sorting of "ls -ltr" under some circumstances. bz3248.

* ssh(1), sshd(8): fix potential integer truncation of (unlikely) timeout values. bz#3250

* ssh(1): make hostbased authentication send the signature algorithm in its SSH2_MSG_USERAUTH_REQUEST packets instead of the key type. This make HostbasedAcceptedAlgorithms do what it is supposed to - filter on signature algorithm and not key type.

Portability
----

* sshd(8): add a number of platform-specific syscalls to the Linux

```
    seccomp-bpf sandbox. bz#3232 bz#3260

 * sshd(8): remove debug message from sigchld handler that could cause
   deadlock on some platforms. bz#3259

 * Sync contrib/ssh-copy-id with upstream.

 * unittests: add a hostname function for systems that don't have it.
   Some systems don't have a hostname command (it's not required by
   POSIX). The do have uname -n (which is), but not all of those have
   it report the FQDN.

Checksums:
==========

 - SHA1 (openssh-8.5.tar.gz) = 04cae43c389fb411227c01219e4eb46e3113f34e
 - SHA256 (openssh-8.5.tar.gz) = 5qB2CgzNG4io4DmChTjHgCWqRWvEOvCKJskLdJCz+SU=

 - SHA1 (openssh-8.5p1.tar.gz) = 72eadcbe313b07b1dd3b693e41d3cd56d354e24e
 - SHA256 (openssh-8.5p1.tar.gz) = 9S8/QdQpqpkY44zyAK8iXM3Y5m8FLaVyhwyJc3ZG7CU=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available from the mirror sites:
https://cdn.openbsd.org/pub/OpenBSD/OpenSSH/RELEASE_KEY.asc

Please note that the OpenPGP key used to sign releases has been
rotated for this release. The new key has been signed by the previous
key to provide continuity.

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com
```

OpenSSH 8.4/8.4p1 (2020-09-27)

```
OpenSSH 8.4 was released on 2020-09-27. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
https://www.openssh.com/donations.html

Future deprecation notice
=========================

It is now possible[1] to perform chosen-prefix attacks against the
SHA-1 algorithm for less than USD$50K. For this reason, we will be
disabling the "ssh-rsa" public key signature algorithm by default in a
near-future release.

This algorithm is unfortunately still used widely despite the
existence of better alternatives, being the only remaining public key
signature algorithm specified by the original SSH RFCs.

The better alternatives include:

 * The RFC8332 RSA SHA-2 signature algorithms rsa-sha2-256/512. These
   algorithms have the advantage of using the same key type as
   "ssh-rsa" but use the safe SHA-2 hash algorithms. These have been
   supported since OpenSSH 7.2 and are already used by default if the
   client and server support them.

 * The ssh-ed25519 signature algorithm. It has been supported in
   OpenSSH since release 6.5.

 * The RFC5656 ECDSA algorithms: ecdsa-sha2-nistp256/384/521. These
   have been supported by OpenSSH since release 5.7.

To check whether a server is using the weak ssh-rsa public key
algorithm, for host authentication, try to connect to it after
removing the ssh-rsa algorithm from ssh(1)'s allowed list:

    ssh -oHostKeyAlgorithms=-ssh-rsa user@host

If the host key verification fails and no other supported host key
types are available, the server software on that host should be
upgraded.

We intend to enable UpdateHostKeys by default in the next OpenSSH
release. This will assist the client by automatically migrating to
better algorithms. Users may consider enabling this option manually.

[1] "SHA-1 is a Shambles: First Chosen-Prefix Collision on SHA-1 and
    Application to the PGP Web of Trust" Leurent, G and Peyrin, T
    (2020) https://eprint.iacr.org/2020/014.pdf

Security
========
```

* ssh-agent(1): restrict ssh-agent from signing web challenges for FIDO/U2F keys.

  When signing messages in ssh-agent using a FIDO key that has an application string that does not start with "ssh:", ensure that the message being signed is one of the forms expected for the SSH protocol (currently public key authentication and sshsig signatures).

  This prevents ssh-agent forwarding on a host that has FIDO keys attached granting the ability for the remote side to sign challenges for web authentication using those keys too.

  Note that the converse case of web browsers signing SSH challenges is already precluded because no web RP can have the "ssh:" prefix in the application string that we require.

* ssh-keygen(1): Enable FIDO 2.1 credProtect extension when generating a FIDO resident key.

  The recent FIDO 2.1 Client to Authenticator Protocol introduced a "credProtect" feature to better protect resident keys. We use this option to require a PIN prior to all operations that may retrieve a resident key from a FIDO token.

Potentially-incompatible changes
================================

This release includes a number of changes that may affect existing configurations:

* For FIDO/U2F support, OpenSSH recommends the use of libfido2 1.5.0 or greater. Older libraries have limited support at the expense of disabling particular features. These include resident keys, PIN-required keys and multiple attached tokens.

* ssh-keygen(1): the format of the attestation information optionally recorded when a FIDO key is generated has changed. It now includes the authenticator data needed to validate attestation signatures.

* The API between OpenSSH and the FIDO token middleware has changed and the SSH_SK_VERSION_MAJOR version has been incremented as a result. Third-party middleware libraries must support the current API version (7) to work with OpenSSH 8.4.

* The portable OpenSSH distribution now requires automake to rebuild the configure script and supporting files. This is not required when simply building portable OpenSSH from a release tar file.

Changes since OpenSSH 8.3
=========================

New features
----

* ssh(1), ssh-keygen(1): support for FIDO keys that require a PIN for each use. These keys may be generated using ssh-keygen using a new "verify-required" option. When a PIN-required key is used, the user will be prompted for a PIN to complete the signature operation.

* sshd(8): authorized_keys now supports a new "verify-required" option to require FIDO signatures assert that the token verified that the user was present before making the signature. The FIDO protocol supports multiple methods for user-verification, but currently OpenSSH only supports PIN verification.

* sshd(8), ssh-keygen(1): add support for verifying FIDO webauthn signatures. Webauthn is a standard for using FIDO keys in web browsers. These signatures are a slightly different format to plain FIDO signatures and thus require explicit support.

* ssh(1): allow some keywords to expand shell-style ${ENV} environment variables. The supported keywords are CertificateFile, ControlPath, IdentityAgent and IdentityFile, plus LocalForward and RemoteForward when used for Unix domain socket paths. bz#3140

* ssh(1), ssh-agent(1): allow some additional control over the use of ssh-askpass via a new $SSH_ASKPASS_REQUIRE environment variable, including forcibly enabling and disabling its use. bz#69

* ssh(1): allow ssh_config(5)'s AddKeysToAgent keyword accept a time limit for keys in addition to its current flag options. Time-limited keys will automatically be removed from ssh-agent after their expiry time has passed.

* scp(1), sftp(1): allow the -A flag to explicitly enable agent forwarding in scp and sftp. The default remains to not forward an agent, even when ssh_config enables it.

* ssh(1): add a '%k' TOKEN that expands to the effective HostKey of the destination. This allows, e.g., keeping host keys in individual files using "UserKnownHostsFile ~/.ssh/known_hosts.d/%k". bz#1654

* ssh(1): add %-TOKEN, environment variable and tilde expansion to the UserKnownHostsFile directive, allowing the path to be completed by the configuration (e.g. bz#1654)

* ssh-keygen(1): allow "ssh-add -d -" to read keys to be deleted from stdin. bz#3180

* sshd(8): improve logging for MaxStartups connection throttling.
  sshd will now log when it starts and stops throttling and periodically
  while in this state. bz#3055

Bugfixes
----

* ssh(1), ssh-keygen(1): better support for multiple attached FIDO
  tokens. In cases where OpenSSH cannot unambiguously determine which
  token to direct a request to, the user is now required to select a
  token by touching it. In cases of operations that require a PIN to
  be verified, this avoids sending the wrong PIN to the wrong token
  and incrementing the token's PIN failure counter (tokens
  effectively erase their keys after too many PIN failures).

* sshd(8): fix Include before Match in sshd_config; bz#3122

* ssh(1): close stdin/out/error when forking after authentication
  completes ("ssh -f ...") bz#3137

* ssh(1), sshd(8): limit the amount of channel input data buffered,
  avoiding peers that advertise large windows but are slow to read
  from causing high memory consumption.

* ssh-agent(1): handle multiple requests sent in a single write() to
  the agent.

* sshd(8): allow sshd_config longer than 256k

* sshd(8): avoid spurious "Unable to load host key" message when sshd
  load a private key but no public counterpart

* ssh(1): prefer the default hostkey algorithm list whenever we have
  a hostkey that matches its best-preference algorithm.

* sshd(1): when ordering the hostkey algorithms to request from a
  server, prefer certificate types if the known_hosts files contain a key
  marked as a @cert-authority; bz#3157

* ssh(1): perform host key fingerprint comparisons for the "Are you
  sure you want to continue connecting (yes/no/[fingerprint])?"
  prompt with case sensitivity.

* sshd(8): ensure that address/masklen mismatches in sshd_config
  yield fatal errors at daemon start time rather than later when
  they are evaluated.

* ssh-keygen(1): ensure that certificate extensions are lexically
  sorted. Previously if the user specified a custom extension then
  the everything would be in order except the custom ones. bz#3198

* ssh(1): also compare username when checking for JumpHost loops.
  bz#3057

* ssh-keygen(1): preserve group/world read permission on known_hosts
  files across runs of "ssh-keygen -Rf /path". The old behaviour was
  to remove all rights for group/other. bz#3146

* ssh-keygen(1): Mention the [-a rounds] flag in the ssh-keygen
  manual page and usage().

* sshd(8): explicitly construct path to ~/.ssh/rc rather than
  relying on it being relative to the current directory, so that it
  can still be found if the shell startup changes its directory.
  bz#3185

* sshd(8): when redirecting sshd's log output to a file, undo this
  redirection after the session child process is forked(). Fixes
  missing log messages when using this feature under some
  circumstances.

* sshd(8): start ClientAliveInterval bookkeeping before first pass
  through select() loop; fixed theoretical case where busy sshd may
  ignore timeouts from client.

* ssh(1): only reset the ServerAliveInterval check when we receive
  traffic from the server and ignore traffic from a port forwarding
  client, preventing a client from keeping a connection alive when
  it should be terminated. bz#2265

* ssh-keygen(1): avoid spurious error message when ssh-keygen
  creates files outside ~/.ssh

* sftp-client(1): fix off-by-one error that caused sftp downloads to
  make one more concurrent request that desired. This prevented using
  sftp(1) in unpipelined request/response mode, which is useful when
  debugging. bz#3054

* ssh(1), sshd(8): handle EINTR in waitfd() and timeout_connect()
  helpers. bz#3071

* ssh(1), ssh-keygen(1): defer creation of ~/.ssh until we attempt to
  write to it so we don't leave an empty .ssh directory when it's not
  needed. bz#3156

* ssh(1), sshd(8): fix multiplier when parsing time specifications
  when handling seconds after other units. bz#3171

```
Portability
-----------

 * sshd(8): always send any PAM account messages. If the PAM account
   stack returns any messages, always send them to the user and not
   just if the check succeeds. bz#2049

 * Implement some backwards compatibility for libfido2 libraries
   older than 1.5.0. Note that use of an older library will result
   in the loss of certain features including resident key support,
   PIN support and support for multiple attached tokens.

 * configure fixes for XCode 12

 * gnome-ssh-askpass3: ensure the "close" button is not focused by
   default for SSH_ASKPASS_PROMPT=none prompts. Avoids space/enter
   accidentally dismissing FIDO touch notifications.

 * gnome-ssh-askpass3: allow some control over textarea colour via
   $GNOME_SSH_ASKPASS_FG_COLOR and $GNOME_SSH_ASKPASS_BG_COLOR
   environment variables.

 * sshd(8): document another PAM spec problem in a frustrated comment

 * sshd(8): support NetBSD's utmpx.ut_ss address field. bz#960

 * Add the ssh-sk-helper binary and its manpage to the RPM spec file

 * Detect the Frankenstein monster of Linux/X32 and allow the sandbox
   to function there. bz#3085

Checksums:
==========

- SHA1 (openssh-8.4.tar.gz) = 71675139df6807f396e6bd92ff8cb9b0356385d8
- SHA256 (openssh-8.4.tar.gz) = JhBgLYkyRge/zQK8ylBSRcOYvrV/tHwQcvVXfExGB70=

- SHA1 (openssh-8.4p1.tar.gz) = 69305059e10a60693ebe6f17731f962c9577535c
- SHA256 (openssh-8.4p1.tar.gz) = WgHSLkB+scBbqKj3xlTTiKE+nyJuTtM704dI2vodKyQ=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available as RELEASE_KEY.asc from
the mirror sites.

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com
```

OpenSSH 8.3/8.3p1 (2020-05-27)

```
OpenSSH 8.3 was released on 2020-05-27. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
https://www.openssh.com/donations.html

Future deprecation notice
=========================

It is now possible[1] to perform chosen-prefix attacks against the
SHA-1 algorithm for less than USD$50K. For this reason, we will be
disabling the "ssh-rsa" public key signature algorithm by default in a
near-future release.

This algorithm is unfortunately still used widely despite the
existence of better alternatives, being the only remaining public key
signature algorithm specified by the original SSH RFCs.

The better alternatives include:

 * The RFC8332 RSA SHA-2 signature algorithms rsa-sha2-256/512. These
   algorithms have the advantage of using the same key type as
   "ssh-rsa" but use the safe SHA-2 hash algorithms. These have been
   supported since OpenSSH 7.2 and are already used by default if the
   client and server support them.

 * The ssh-ed25519 signature algorithm. It has been supported in
   OpenSSH since release 6.5.

 * The RFC5656 ECDSA algorithms: ecdsa-sha2-nistp256/384/521. These
   have been supported by OpenSSH since release 5.7.

To check whether a server is using the weak ssh-rsa public key
algorithm, for host authentication, try to connect to it after
removing the ssh-rsa algorithm from ssh(1)'s allowed list:

    ssh -oHostKeyAlgorithms=-ssh-rsa user@host
```

If the host key verification fails and no other supported host key types are available, the server software on that host should be upgraded.

A future release of OpenSSH will enable UpdateHostKeys by default to allow the client to automatically migrate to better algorithms. Users may consider enabling this option manually. Vendors of devices that implement the SSH protocol should ensure that they support the new signature algorithms for RSA keys.

[1] "SHA-1 is a Shambles: First Chosen-Prefix Collision on SHA-1 and Application to the PGP Web of Trust" Leurent, G and Peyrin, T (2020) https://eprint.iacr.org/2020/014.pdf

Security
========

 * scp(1): when receiving files, scp(1) could be become desynchronised if a utimes(2) system call failed. This could allow file contents to be interpreted as file metadata and thereby permit an adversary to craft a file system that, when copied with scp(1) in a configuration that caused utimes(2) to fail (e.g. under a SELinux policy or syscall sandbox), transferred different file names and contents to the actual file system layout.

   Exploitation of this is not likely as utimes(2) does not fail under normal circumstances. Successful exploitation is not silent - the output of scp(1) would show transfer errors followed by the actual file(s) that were received.

   Finally, filenames returned from the peer are (since openssh-8.0) matched against the user's requested destination, thereby disallowing a successful exploit from writing files outside the user's selected target glob (or directory, in the case of a recursive transfer). This ensures that this attack can achieve no more than a hostile peer is already able to achieve within the scp protocol.

Potentially-incompatible changes
================================

This release includes a number of changes that may affect existing configurations:

 * sftp(1): reject an argument of "-1" in the same way as ssh(1) and scp(1) do instead of accepting and silently ignoring it.

Changes since OpenSSH 8.2
=========================

The focus of this release is bug fixing.

New Features
----

 * sshd(8): make IgnoreRhosts a tri-state option: "yes" to ignore rhosts/shosts, "no" allow rhosts/shosts or (new) "shosts-only" to allow .shosts files but not .rhosts.

 * sshd(8): allow the IgnoreRhosts directive to appear anywhere in a sshd_config, not just before any Match blocks; bz3148

 * ssh(1): add %TOKEN percent expansion for the LocalFoward and RemoteForward keywords when used for Unix domain socket forwarding. bz#3014

 * all: allow loading public keys from the unencrypted envelope of a private key file if no corresponding public key file is present.

 * ssh(1), sshd(8): prefer to use chacha20 from libcrypto where possible instead of the (slower) portable C implementation included in OpenSSH.

 * ssh-keygen(1): add ability to dump the contents of a binary key revocation list via "ssh-keygen -lQf /path" bz#3132

Bugfixes
----

 * ssh(1): fix IdentitiesOnly=yes to also apply to keys loaded from a PKCS11Provider; bz#3141

 * ssh-keygen(1): avoid NULL dereference when trying to convert an invalid RFC4716 private key.

 * scp(1): when performing remote-to-remote copies using "scp -3", start the second ssh(1) channel with BatchMode=yes enabled to avoid confusing and non-deterministic ordering of prompts.

 * ssh(1), ssh-keygen(1): when signing a challenge using a FIDO token, perform hashing of the message to be signed in the middleware layer rather than in OpenSSH code. This permits the use of security key middlewares that perform the hashing implicitly, such as Windows Hello.

 * ssh(1): fix incorrect error message for "too many known hosts files." bz#3149

```
 * ssh(1): make failures when establishing "Tunnel" forwarding
   terminate the connection when ExitOnForwardFailure is enabled;
   bz#3116

 * ssh-keygen(1): fix printing of fingerprints on private keys and add
   a regression test for same.

 * sshd(8): document order of checking AuthorizedKeysFile (first) and
   AuthorizedKeysCommand (subsequently, if the file doesn't match);
   bz#3134

 * sshd(8): document that /etc/hosts.equiv and /etc/shosts.equiv are
   not considered for HostbasedAuthentication when the target user is
   root; bz#3148

 * ssh(1), ssh-keygen(1): fix NULL dereference in private certificate
   key parsing (oss-fuzz #20074).

 * ssh(1), sshd(8): more consistency between sets of %TOKENS are
   accepted in various configuration options.

 * ssh(1), ssh-keygen(1): improve error messages for some common
   PKCS#11 C_Login failure cases; bz#3130

 * ssh(1), sshd(8): make error messages for problems during SSH banner
   exchange consistent with other SSH transport-layer error messages
   and ensure they include the relevant IP addresses bz#3129

 * various: fix a number of spelling errors in comments and debug/error
   messages

 * ssh-keygen(1), ssh-add(1): when downloading FIDO2 resident keys
   from a token, don't prompt for a PIN until the token has told us
   that it needs one. Avoids double-prompting on devices that
   implement on-device authentication.

 * sshd(8), ssh-keygen(1): no-touch-required FIDO certificate option
   should be an extension, not a critical option.

 * ssh(1), ssh-keygen(1), ssh-add(1): offer a better error message
   when trying to use a FIDO key function and SecurityKeyProvider is
   empty.

 * ssh-add(1), ssh-agent(8): ensure that a key lifetime fits within
   the values allowed by the wire format (u32). Prevents integer
   wraparound of the timeout values. bz#3119

 * ssh(1): detect and prevent trivial configuration loops when using
    ProxyJump. bz#3057.

Portability
-----------

 * Detect systems where signals flagged with SA_RESTART will interrupt
   select(2). POSIX permits implementations to choose whether
   select(2) will return when interrupted with a SA_RESTART-flagged
   signal, but OpenSSH requires interrupting behaviour.

 * Several compilation fixes for HP/UX and AIX.

 * On platforms that do not support setting process-wide routing
   domains (all excepting OpenBSD at present), fail to accept a
   configuration attempts to set one at process start time rather than
   fatally erroring at run time. bz#3126

 * Improve detection of egrep (used in regression tests) on platforms
   that offer a poor default one (e.g. Solaris).

 * A number of shell portability fixes for the regression tests.

 * Fix theoretical infinite loop in the glob(3) replacement
   implementation.

 * Fix seccomp sandbox compilation problems for some Linux
   configurations bz#3085

 * Improved detection of libfido2 and some compilation fixes for some
   configurations when --with-security-key-builtin is selected.

Checksums:
==========

 - SHA1 (openssh-8.3.tar.gz) = 46c63b7ddbe46a0666222f7988c993866c31fcca
 - SHA256 (openssh-8.3.tar.gz) = M6CnZ+duGs4bzDio8hQNLwyLQChV+3wkUEO8HWLV35c=

 - SHA1 (/openssh-8.3p1.tar.gz) = 04c7adb9986f16746588db8988b910530c589819
 - SHA256 (openssh-8.3p1.tar.gz) = 8r774Ecv5+t10jNA6xdTHLazqsJAdeIGa0H4FOEjh7I=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available as RELEASE_KEY.asc from
the mirror sites.

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
```

```
Security bugs should be reported directly to openssh@openssh.com
```

OpenSSH 8.2/8.2p1 (2020-02-14)

```
OpenSSH 8.2 was released on 2020-02-14. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
https://www.openssh.com/donations.html

Future deprecation notice
=========================

It is now possible[1] to perform chosen-prefix attacks against the
SHA-1 hash algorithm for less than USD$50K. For this reason, we will
be disabling the "ssh-rsa" public key signature algorithm that depends
on SHA-1 by default in a near-future release.

This algorithm is unfortunately still used widely despite the
existence of better alternatives, being the only remaining public key
signature algorithm specified by the original SSH RFCs.

The better alternatives include:

 * The RFC8332 RSA SHA-2 signature algorithms rsa-sha2-256/512. These
   algorithms have the advantage of using the same key type as
   "ssh-rsa" but use the safe SHA-2 hash algorithms. These have been
   supported since OpenSSH 7.2 and are already used by default if the
   client and server support them.

 * The ssh-ed25519 signature algorithm. It has been supported in
   OpenSSH since release 6.5.

 * The RFC5656 ECDSA algorithms: ecdsa-sha2-nistp256/384/521. These
   have been supported by OpenSSH since release 5.7.

To check whether a server is using the weak ssh-rsa public key
algorithm for host authentication, try to connect to it after
removing the ssh-rsa algorithm from ssh(1)'s allowed list:

    ssh -oHostKeyAlgorithms=-ssh-rsa user@host

If the host key verification fails and no other supported host key
types are available, the server software on that host should be
upgraded.

A future release of OpenSSH will enable UpdateHostKeys by default
to allow the client to automatically migrate to better algorithms.
Users may consider enabling this option manually.

[1] "SHA-1 is a Shambles: First Chosen-Prefix Collision on SHA-1 and
    Application to the PGP Web of Trust" Leurent, G and Peyrin, T
    (2020) https://eprint.iacr.org/2020/014.pdf

Security
========

 * ssh(1), sshd(8), ssh-keygen(1): this release removes the "ssh-rsa"
   (RSA/SHA1) algorithm from those accepted for certificate signatures
   (i.e. the client and server CASignatureAlgorithms option) and will
   use the rsa-sha2-512 signature algorithm by default when the
   ssh-keygen(1) CA signs new certificates.

   Certificates are at special risk to the aforementioned SHA1
   collision vulnerability as an attacker has effectively unlimited
   time in which to craft a collision that yields them a valid
   certificate, far more than the relatively brief LoginGraceTime
   window that they have to forge a host key signature.

   The OpenSSH certificate format includes a CA-specified (typically
   random) nonce value near the start of the certificate that should
   make exploitation of chosen-prefix collisions in this context
   challenging, as the attacker does not have full control over the
   prefix that actually gets signed. Nonetheless, SHA1 is now a
   demonstrably broken algorithm and futher improvements in attacks
   are highly likely.

   OpenSSH releases prior to 7.2 do not support the newer RSA/SHA2
   algorithms and will refuse to accept certificates signed by an
   OpenSSH 8.2+ CA using RSA keys unless the unsafe algorithm is
   explicitly selected during signing ("ssh-keygen -t ssh-rsa").
   Older clients/servers may use another CA key type such as
   ssh-ed25519 (supported since OpenSSH 6.5) or one of the
   ecdsa-sha2-nistp256/384/521 types (supported since OpenSSH 5.7)
   instead if they cannot be upgraded.

Potentially-incompatible changes
================================

This release includes a number of changes that may affect existing
```

configurations:

 * ssh(1), sshd(8): the above removal of "ssh-rsa" from the accepted
   CASignatureAlgorithms list.

 * ssh(1), sshd(8): this release removes diffie-hellman-group14-sha1
   from the default key exchange proposal for both the client and
   server.

 * ssh-keygen(1): the command-line options related to the generation
   and screening of safe prime numbers used by the
   diffie-hellman-group-exchange-* key exchange algorithms have
   changed. Most options have been folded under the -O flag.

 * sshd(8): the sshd listener process title visible to ps(1) has
   changed to include information about the number of connections that
   are currently attempting authentication and the limits configured
   by MaxStartups.

 * ssh-sk-helper(8): this is a new binary. It is used by the FIDO/U2F
   support to provide address-space isolation for token middleware
   libraries (including the internal one). It needs to be installed
   in the expected path, typically under /usr/libexec or similar.

Changes since OpenSSH 8.1
=========================

This release contains some significant new features.

FIDO/U2F Support

This release adds support for FIDO/U2F hardware authenticators to
OpenSSH. U2F/FIDO are open standards for inexpensive two-factor
authentication hardware that are widely used for website
authentication. In OpenSSH FIDO devices are supported by new public
key types "ecdsa-sk" and "ed25519-sk", along with corresponding
certificate types.

ssh-keygen(1) may be used to generate a FIDO token-backed key, after
which they may be used much like any other key type supported by
OpenSSH, so long as the hardware token is attached when the keys are
used. FIDO tokens also generally require the user explicitly authorise
operations by touching or tapping them.

Generating a FIDO key requires the token be attached, and will usually
require the user tap the token to confirm the operation:

```
  $ ssh-keygen -t ecdsa-sk -f ~/.ssh/id_ecdsa_sk
  Generating public/private ecdsa-sk key pair.
  You may need to touch your security key to authorize key generation.
  Enter file in which to save the key (/home/djm/.ssh/id_ecdsa_sk):
  Enter passphrase (empty for no passphrase):
  Enter same passphrase again:
  Your identification has been saved in /home/djm/.ssh/id_ecdsa_sk
  Your public key has been saved in /home/djm/.ssh/id_ecdsa_sk.pub
```

This will yield a public and private key-pair. The private key file
should be useless to an attacker who does not have access to the
physical token. After generation, this key may be used like any other
supported key in OpenSSH and may be listed in authorized_keys, added
to ssh-agent(1), etc. The only additional stipulation is that the FIDO
token that the key belongs to must be attached when the key is used.

FIDO tokens are most commonly connected via USB but may be attached
via other means such as Bluetooth or NFC. In OpenSSH, communication
with the token is managed via a middleware library, specified by the
SecurityKeyProvider directive in ssh/sshd_config(5) or the
$SSH_SK_PROVIDER environment variable for ssh-keygen(1) and
ssh-add(1). The API for this middleware is documented in the sk-api.h
and PROTOCOL.u2f files in the source distribution.

OpenSSH includes a middleware ("SecurityKeyProvider=internal") with
support for USB tokens. It is automatically enabled in OpenBSD and may
be enabled in portable OpenSSH via the configure flag
--with-security-key-builtin. If the internal middleware is enabled
then it is automatically used by default. This internal middleware
requires that libfido2 (https://github.com/Yubico/libfido2)and its
dependencies be installed. We recommend that packagers of portable
OpenSSH enable the built-in middleware, as it provides the
lowest-friction experience for users.

Note: FIDO/U2F tokens are required to implement the ECDSA-P256
"ecdsa-sk" key type, but hardware support for Ed25519 "ed25519-sk" is
less common. Similarly, not all hardware tokens support some of the
optional features such as resident keys.

The protocol-level changes to support FIDO/U2F keys in SSH are
documented in the PROTOCOL.u2f file in the OpenSSH source
distribution.

There are a number of supporting changes to this feature:

 * ssh-keygen(1): add a "no-touch-required" option when generating
   FIDO-hosted keys, that disables their default behaviour of
   requiring a physical touch/tap on the token during authentication.
   Note: not all tokens support disabling the touch requirement.

* sshd(8): add a sshd_config PubkeyAuthOptions directive that collects miscellaneous public key authentication-related options for sshd(8). At present it supports only a single option "no-touch-required". This causes sshd to skip its default check for FIDO/U2F keys that the signature was authorised by a touch or press event on the token hardware.

* ssh(1), sshd(8), ssh-keygen(1): add a "no-touch-required" option for authorized_keys and a similar extension for certificates. This option disables the default requirement that FIDO key signatures attest that the user touched their key to authorize them, mirroring the similar PubkeyAuthOptions sshd_config option.

* ssh-keygen(1): add support for the writing the FIDO attestation information that is returned when new keys are generated via the "-O write-attestation=/path" option. FIDO attestation certificates may be used to verify that a FIDO key is hosted in trusted hardware. OpenSSH does not currently make use of this information, beyond optionally writing it to disk.

FIDO2 resident keys

FIDO/U2F OpenSSH keys consist of two parts: a "key handle" part stored in the private key file on disk, and a per-device private key that is unique to each FIDO/U2F token and that cannot be exported from the token hardware. These are combined by the hardware at authentication time to derive the real key that is used to sign authentication challenges.

For tokens that are required to move between computers, it can be cumbersome to have to move the private key file first. To avoid this requirement, tokens implementing the newer FIDO2 standard support "resident keys", where it is possible to effectively retrieve the key handle part of the key from the hardware.

OpenSSH supports this feature, allowing resident keys to be generated using the ssh-keygen(1) "-O resident" flag. This will produce a public/private key pair as usual, but it will be possible to retrieve the private key part from the token later. This may be done using "ssh-keygen -K", which will download all available resident keys from the tokens attached to the host and write public/private key files for them. It is also possible to download and add resident keys directly to ssh-agent(1) without writing files to the file-system using "ssh-add -K".

Resident keys are indexed on the token by the application string and user ID. By default, OpenSSH uses an application string of "ssh:" and an empty user ID. If multiple resident keys on a single token are desired then it may be necessary to override one or both of these defaults using the ssh-keygen(1) "-O application=" or "-O user=" options. Note: OpenSSH will only download and use resident keys whose application string begins with "ssh:"

Storing both parts of a key on a FIDO token increases the likelihood of an attacker being able to use a stolen token device. For this reason, tokens should enforce PIN authentication before allowing download of keys, and users should set a PIN on their tokens before creating any resident keys.

Other New Features
----

* sshd(8): add an Include sshd_config keyword that allows including additional configuration files via glob(3) patterns. bz2468

* ssh(1)/sshd(8): make the LE (low effort) DSCP code point available via the IPQoS directive; bz2986,

* ssh(1): when AddKeysToAgent=yes is set and the key contains no comment, add the key to the agent with the key's path as the comment. bz2564

* ssh-keygen(1), ssh-agent(1): expose PKCS#11 key labels and X.509 subjects as key comments, rather than simply listing the PKCS#11 provider library path. PR138

* ssh-keygen(1): allow PEM export of DSA and ECDSA keys; bz3091

* ssh(1), sshd(8): make zlib compile-time optional, available via the Makefile.inc ZLIB flag on OpenBSD or via the --with-zlib configure option for OpenSSH portable.

* sshd(8): when clients get denied by MaxStartups, send a notification prior to the SSH2 protocol banner according to RFC4253 section 4.2.

* ssh(1), ssh-agent(1): when invoking the $SSH_ASKPASS prompt program, pass a hint to the program to describe the type of desired prompt. The possible values are "confirm" (indicating that a yes/no confirmation dialog with no text entry should be shown), "none" (to indicate an informational message only), or blank for the original ssh-askpass behaviour of requesting a password/phrase.

* ssh(1): allow forwarding a different agent socket to the path specified by $SSH_AUTH_SOCK, by extending the existing ForwardAgent option to accepting an explicit path or the name of an environment

variable in addition to yes/no.

* ssh-keygen(1): add a new signature operations "find-principals" to look up the principal associated with a signature from an allowed-signers file.

* sshd(8): expose the number of currently-authenticating connections along with the MaxStartups limit in the process title visible to "ps".

Bugfixes
----

* sshd(8): make ClientAliveCountMax=0 have sensible semantics: it will now disable connection killing entirely rather than the current behaviour of instantly killing the connection after the first liveness test regardless of success. bz2627

* sshd(8): clarify order of AllowUsers / DenyUsers vs AllowGroups / DenyGroups in the sshd(8) manual page. bz1690

* sshd(8): better describe HashKnownHosts in the manual page. bz2560

* sshd(8): clarify that that permitopen=/PermitOpen do no name or address translation in the manual page. bz3099

* sshd(8): allow the UpdateHostKeys feature to function when multiple known_hosts files are in use. When updating host keys, ssh will now search subsequent known_hosts files, but will add updated host keys to the first specified file only. bz2738

* All: replace all calls to signal(2) with a wrapper around sigaction(2). This wrapper blocks all other signals during the handler preventing races between handlers, and sets SA_RESTART which should reduce the potential for short read/write operations.

* sftp(1): fix a race condition in the SIGCHILD handler that could turn in to a kill(-1); bz3084

* sshd(8): fix a case where valid (but extremely large) SSH channel IDs were being incorrectly rejected. bz3098

* ssh(1): when checking host key fingerprints as answers to new hostkey prompts, ignore whitespace surrounding the fingerprint itself.

* All: wait for file descriptors to be readable or writeable during non-blocking connect, not just readable. Prevents a timeout when the server doesn't immediately send a banner (e.g. multiplexers like sslh)

* sshd_config(5): document the sntrup4591761x25519-sha512@tinyssh.org key exchange algorithm. PR#151

Portability
----

* sshd(8): multiple adjustments to the Linux seccomp sandbox:
  - Non-fatally deny IPC syscalls in sandbox
  - Allow clock_gettime64() in sandbox (MIPS / glibc >= 2.31)
  - Allow clock_nanosleep_time64 in sandbox (ARM) bz3100
  - Allow clock_nanosleep() in sandbox (recent glibc) bz3093

* Explicit check for memmem declaration and fix up declaration if the system headers lack it. bz3102

Checksums:
==========

- SHA1 (openssh-8.2.tar.gz) = 0daae2a8c47c489a8784f2c38c4b39e6159ba678
- SHA256 (openssh-8.2.tar.gz) = +UmInEIoHJqYqWneMb/kgRbLcq8WDCo7+ooYcjzW4jg=

- SHA1 (openssh-8.2p1.tar.gz) = d1ab35a93507321c5db885e02d41ce1414f0507c
- SHA256 (openssh-8.2p1.tar.gz) = Q5JRUebPbO4UUBkMDpr03Da0HBJzdhnt/4vOvf9k5nE=

Note: the openssh-8.2 tarball for OpenBSD that was initially released advertised an incorrect version for "ssh -V" and the sshd server banner. The above tarball replace the incorrect release, which has been renamed to openssh-8.2.tar.gz.incorrect. These are the checksums for the original, incorrect tarball:

- SHA1 (openssh-8.2.tar.gz) = 77584c22fbb89269398acdf53c1e554400584ba8
- SHA256 (openssh-8.2.tar.gz) = UttLaaSYXVK1O65cYvyQzyQ5sCfuJ4Lwrs8zNsPrluQ=

Please note that the SHA256 signatures are base64 encoded and not hexadecimal (which is the default for most checksum tools). The PGP key used to sign the releases is available as RELEASE_KEY.asc from the mirror sites.

Reporting Bugs:
===============

- Please read https://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

---

OpenSSH 8.1/8.1p1 (2019-10-09)

OpenSSH 8.1 was released on 2019-10-09. It is available from the mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their continued support of the project, especially those who contributed code or patches, reported bugs, tested snapshots or donated to the project. More information on donations may be found at: http://www.openssh.com/donations.html

Security

 * ssh(1), sshd(8), ssh-add(1), ssh-keygen(1): an exploitable integer overflow bug was found in the private key parsing code for the XMSS key type. This key type is still experimental and support for it is not compiled by default. No user-facing autoconf option exists in portable OpenSSH to enable it. This bug was found by Adam Zabrocki and reported via SecuriTeam's SSD program.

 * ssh(1), sshd(8), ssh-agent(1): add protection for private keys at rest in RAM against speculation and memory side-channel attacks like Spectre, Meltdown and Rambleed. This release encrypts private keys when they are not in use with a symmetric key that is derived from a relatively large "prekey" consisting of random data (currently 16KB).

Potentially-incompatible changes

This release includes a number of changes that may affect existing configurations:

 * ssh-keygen(1): when acting as a CA and signing certificates with an RSA key, default to using the rsa-sha2-512 signature algorithm. Certificates signed by RSA keys will therefore be incompatible with OpenSSH versions prior to 7.2 unless the default is overridden (using "ssh-keygen -t ssh-rsa -s ...").

Changes since OpenSSH 8.0

This release is focused on bug-fixing.

New Features

 * ssh(1): Allow %n to be expanded in ProxyCommand strings

 * ssh(1), sshd(8): Allow prepending a list of algorithms to the default set by starting the list with the '^' character, E.g. "HostKeyAlgorithms ^ssh-ed25519"

 * ssh-keygen(1): add an experimental lightweight signature and verification ability. Signatures may be made using regular ssh keys held on disk or stored in a ssh-agent and verified against an authorized_keys-like list of allowed keys. Signatures embed a namespace that prevents confusion and attacks between different usage domains (e.g. files vs email).

 * ssh-keygen(1): print key comment when extracting public key from a private key. bz#3052

 * ssh-keygen(1): accept the verbose flag when searching for host keys in known hosts (i.e. "ssh-keygen -vF host") to print the matching host's random-art signature too. bz#3003

 * All: support PKCS8 as an optional format for storage of private keys to disk. The OpenSSH native key format remains the default, but PKCS8 is a superior format to PEM if interoperability with non-OpenSSH software is required, as it may use a less insecure key derivation function than PEM's.

Bugfixes

 * ssh(1): if a PKCS#11 token returns no keys then try to login and refetch them. Based on patch from Jakub Jelen; bz#2430

 * ssh(1): produce a useful error message if the user's shell is set incorrectly during "match exec" processing. bz#2791

 * sftp(1): allow the maximum uint32 value for the argument passed to -b which allows better error messages from later validation. bz#3050

 * ssh(1): avoid pledge sandbox violations in some combinations of remote forwarding, connection multiplexing and ControlMaster.

 * ssh-keyscan(1): include SHA2-variant RSA key algorithms in KEX proposal; allows ssh-keyscan to harvest keys from servers that disable old SHA1 ssh-rsa. bz#3029

 * sftp(1): print explicit "not modified" message if a file was requested for resumed download but was considered already complete.

bz#2978

* sftp(1): fix a typo and make <esc><right> move right to the closest end of a word just like <esc><left> moves left to the closest beginning of a word.

* sshd(8): cap the number of permitopen/permitlisten directives allowed to appear on a single authorized_keys line.

* All: fix a number of memory leaks (one-off or on exit paths).

* Regression tests: a number of fixes and improvements, including fixes to the interop tests, adding the ability to run most tests on builds that disable OpenSSL support, better support for running tests under Valgrind and a number of bug-fixes.

* ssh(1), sshd(8): check for convtime() refusing to accept times that resolve to LONG_MAX Reported by Kirk Wolf bz2977

* ssh(1): slightly more instructive error message when the user specifies multiple -J options on the command-line. bz3015

* ssh-agent(1): process agent requests for RSA certificate private keys using correct signature algorithm when requested. bz3016

* sftp(1): check for user@host when parsing sftp target. This allows user@[1.2.3.4] to work without a path. bz#2999

* sshd(8): enlarge format buffer size for certificate serial number so the log message can record any 64-bit integer without truncation. bz#3012

* sshd(8): for PermitOpen violations add the remote host and port to be able to more easily ascertain the source of the request. Add the same logging for PermitListen violations which where not previously logged at all.

* scp(1), sftp(1): use the correct POSIX format style for left justification for the transfer progress meter. bz#3002

* sshd(8) when examining a configuration using sshd -T, assume any attribute not provided by -C does not match, which allows it to work when sshd_config contains a Match directive with or without -C. bz#2858

* ssh(1), ssh-keygen(1): downgrade PKCS#11 "provider returned no slots" warning from log level error to debug. This is common when attempting to enumerate keys on smartcard readers with no cards plugged in. bz#3058

* ssh(1), ssh-keygen(1): do not unconditionally log in to PKCS#11 tokens. Avoids spurious PIN prompts for keys not selected for authentication in ssh(1) and when listing public keys available in a token using ssh-keygen(1). bz#3006

Portability
----

* ssh(1): fix SIGWINCH delivery of Solaris for multiplexed sessions bz#3030

* ssh(1), sshd(8): fix typo that prevented detection of Linux VRF

* sshd(8): add no-op implementation of pam_putenv to avoid build breakage on platforms where the PAM implementation lacks this function (e.g. HP-UX). bz#3008

* sftp-server(8): fix Solaris privilege sandbox from preventing the legacy sftp rename operation from working (was refusing to allow hard links to files owned by other users). bz#3036

* All: add a proc_pidinfo()-based closefrom() for OS X to avoid the need to brute-force close all high-numbered file descriptors. bz#3049

* sshd(8): in the Linux seccomp-bpf sandbox, allow mprotect(2) with PROT_(READ|WRITE|NONE) only. This syscall is used by some hardened heap allocators. Github PR#142

* sshd(8): in the Linux seccomp-bpf sandbox, allow the s390-specific ioctl for ECC hardware support.

* All: use "doc" man page format if the mandoc(1) tool is present on the system. Previously configure would not select the "doc" man page format if mandoc was present but nroff was not.

* sshd(8): don't install duplicate STREAMS modules on Solaris; check if STREAMS modules are already installed on a pty before installing since when compiling with XPG>=4 they will likely be installed already. Prevents hangs and duplicate lines on the terminal. bz#2945 and bz#2998,

Checksums:
==========

- SHA1 (openssh-8.1.tar.gz) = bf7b0c65a7c0afa5ba9c787f345b8a24fa459add
- SHA256 (openssh-8.1.tar.gz) = vamkKxZTFfgxQXSxGeJ1vbuot0H3Vx9bNBgrvChSrFg=

```
 - SHA1 (openssh-8.1p1.tar.gz) = c44b96094869f177735ae053d92bd5fcab1319de
 - SHA256 (openssh-8.1p1.tar.gz) = AvXb7zg10HU1VvlzzVe0wZtrH2zSTANEXiOsd8obk/8=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available as RELEASE_KEY.asc from
the mirror sites.

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com
```

OpenSSH 8.0/8.0p1 (2019-04-17)

```
OpenSSH 8.0 was released on 2019-04-17. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
http://www.openssh.com/donations.html

Security
========

This release contains mitigation for a weakness in the scp(1) tool
and protocol (CVE-2019-6111): when copying files from a remote system
to a local directory, scp(1) did not verify that the filenames that
the server sent matched those requested by the client. This could
allow a hostile server to create or clobber unexpected local files
with attacker-controlled content.

This release adds client-side checking that the filenames sent from
the server match the command-line request,

The scp protocol is outdated, inflexible and not readily fixed. We
recommend the use of more modern protocols like sftp and rsync for
file transfer instead.

Potentially-incompatible changes
================================

This release includes a number of changes that may affect existing
configurations:

 * scp(1): Relating to the above changes to scp(1); the scp protocol
   relies on the remote shell for wildcard expansion, so there is no
   infallible way for the client's wildcard matching to perfectly
   reflect the server's. If there is a difference between client and
   server wildcard expansion, the client may refuse files from the
   server. For this reason, we have provided a new "-T" flag to scp
   that disables these client-side checks at the risk of
   reintroducing the attack described above.

 * sshd(8): Remove support for obsolete "host/port" syntax. Slash-
   separated host/port was added in 2001 as an alternative to
   host:port syntax for the benefit of IPv6 users. These days there
   are establised standards for this like [::1]:22 and the slash
   syntax is easily mistaken for CIDR notation, which OpenSSH
   supports for some things. Remove the slash notation from
   ListenAddress and PermitOpen; bz#2335

Changes since OpenSSH 7.9
=========================

This release is focused on new features and internal refactoring.

New Features
------------

 * ssh(1), ssh-agent(1), ssh-add(1): Add support for ECDSA keys in
   PKCS#11 tokens.

 * ssh(1), sshd(8): Add experimental quantum-computing resistant
   key exchange method, based on a combination of Streamlined NTRU
   Prime 4591^761 and X25519.

 * ssh-keygen(1): Increase the default RSA key size to 3072 bits,
   following NIST Special Publication 800-57's guidance for a
   128-bit equivalent symmetric security level.

 * ssh(1): Allow "PKCS11Provider=none" to override later instances of
   the PKCS11Provider directive in ssh_config; bz#2974

 * sshd(8): Add a log message for situations where a connection is
   dropped for attempting to run a command but a sshd_config
   ForceCommand=internal-sftp restriction is in effect; bz#2960

 * ssh(1): When prompting whether to record a new host key, accept
   the key fingerprint as a synonym for "yes". This allows the user
```

to paste a fingerprint obtained out of band at the prompt and have the client do the comparison for you.

* ssh-keygen(1): When signing multiple certificates on a single command-line invocation, allow automatically incrementing the certificate serial number.

* scp(1), sftp(1): Accept -J option as an alias to ProxyJump on the scp and sftp command-lines.

* ssh-agent(1), ssh-pkcs11-helper(8), ssh-add(1): Accept "-v" command-line flags to increase the verbosity of output; pass verbose flags though to subprocesses, such as ssh-pkcs11-helper started from ssh-agent.

* ssh-add(1): Add a "-T" option to allowing testing whether keys in an agent are usable by performing a signature and a verification.

* sftp-server(8): Add a "lsetstat@openssh.com" protocol extension that replicates the functionality of the existing SSH2_FXP_SETSTAT operation but does not follow symlinks. bz#2067

* sftp(1): Add "-h" flag to chown/chgrp/chmod commands to request they do not follow symlinks.

* sshd(8): Expose $SSH_CONNECTION in the PAM environment. This makes the connection 4-tuple available to PAM modules that wish to use it in decision-making. bz#2741

* sshd(8): Add a ssh_config "Match final" predicate Matches in same pass as "Match canonical" but doesn't require hostname canonicalisation be enabled. bz#2906

* sftp(1): Support a prefix of '@' to suppress echo of sftp batch commands; bz#2926

* ssh-keygen(1): When printing certificate contents using "ssh-keygen -Lf /path/certificate", include the algorithm that the CA used to sign the cert.

Bugfixes
----

* sshd(8): Fix authentication failures when sshd_config contains "AuthenticationMethods any" inside a Match block that overrides a more restrictive default.

* sshd(8): Avoid sending duplicate keepalives when ClientAliveCount is enabled.

* sshd(8): Fix two race conditions related to SIGHUP daemon restart. Remnant file descriptors in recently-forked child processes could block the parent sshd's attempt to listen(2) to the configured addresses. Also, the restarting parent sshd could exit before any child processes that were awaiting their re-execution state had completed reading it, leaving them in a fallback path.

* ssh(1): Fix stdout potentially being redirected to /dev/null when ProxyCommand=- was in use.

* sshd(8): Avoid sending SIGPIPE to child processes if they attempt to write to stderr after their parent processes have exited; bz#2071

* ssh(1): Fix bad interaction between the ssh_config ConnectTimeout and ConnectionAttempts directives - connection attempts after the first were ignoring the requested timeout; bz#2918

* ssh-keyscan(1): Return a non-zero exit status if no keys were found; bz#2903

* scp(1): Sanitize scp filenames to allow UTF-8 characters without terminal control sequences; bz#2434

* sshd(8): Fix confusion between ClientAliveInterval and time-based RekeyLimit that could cause connections to be incorrectly closed. bz#2757

* ssh(1), ssh-add(1): Correct some bugs in PKCS#11 token PIN handling at initial token login. The attempt to read the PIN could be skipped in some cases, particularly on devices with integrated PIN readers. This would lead to an inability to retrieve keys from these tokens. bz#2652

* ssh(1), ssh-add(1): Support keys on PKCS#11 tokens that set the CKA_ALWAYS_AUTHENTICATE flag by requring a fresh login after the C_SignInit operation. bz#2638

* ssh(1): Improve documentation for ProxyJump/-J, clarifying that local configuration does not apply to jump hosts.

* ssh-keygen(1): Clarify manual - ssh-keygen -e only writes public keys, not private.

* ssh(1), sshd(8): be more strict in processing protocol banners, allowing \r characters only immediately before \n.

* Various: fix a number of memory leaks, including bz#2942 and

bz#2938

 * scp(1), sftp(1): fix calculation of initial bandwidth limits.
   Account for bytes written before the timer starts and adjust the
   schedule on which recalculations are performed. Avoids an initial
   burst of traffic and yields more accurate bandwidth limits;
   bz#2927

 * sshd(8): Only consider the ext-info-c extension during the initial
   key eschange. It shouldn't be sent in subsequent ones, but if it
   is present we should ignore it. This prevents sshd from sending a
   SSH_MSG_EXT_INFO for REKEX for buggy these clients. bz#2929

 * ssh-keygen(1): Clarify manual that ssh-keygen -F (find host in
   authorized_keys) and -R (remove host from authorized_keys) options
   may accept either a bare hostname or a [hostname]:port combo.
   bz#2935

 * ssh(1): Don't attempt to connect to empty SSH_AUTH_SOCK; bz#2936

 * sshd(8): Silence error messages when sshd fails to load some of
   the default host keys. Failure to load an explicitly-configured
   hostkey is still an error, and failure to load any host key is
   still fatal. pr/103

 * ssh(1): Redirect stderr of ProxyCommands to /dev/null when ssh is
   started with ControlPersist; prevents random ProxyCommand output
   from interfering with session output.

 * ssh(1): The ssh client was keeping a redundant ssh-agent socket
   (leftover from authentication) around for the life of the
   connection; bz#2912

 * sshd(8): Fix bug in HostbasedAcceptedKeyTypes and
   PubkeyAcceptedKeyTypes options. If only RSA-SHA2 siganture types
   were specified, then authentication would always fail for RSA keys
   as the monitor checks only the base key (not the signature
   algorithm) type against *AcceptedKeyTypes. bz#2746

 * ssh(1): Request correct signature types from ssh-agent when
   certificate keys and RSA-SHA2 signatures are in use.

Portability
----

 * sshd(8): On Cygwin, run as SYSTEM where possible, using S4U for
   token creation if it supports MsV1_0 S4U Logon.

 * sshd(8): On Cygwin, use custom user/group matching code that
   respects the OS' behaviour of case-insensitive matching.

 * sshd(8): Don't set $MAIL if UsePAM=yes as PAM typically specifies
   the user environment if it's enabled; bz#2937

 * sshd(8) Cygwin: Change service name to cygsshd to avoid collision
   with Microsoft's OpenSSH port.

 * Allow building against OpenSSL -dev (3.x)

 * Fix a number of build problems against version configurations and
   versions of OpenSSL. Including bz#2931 and bz#2921

 * Improve warnings in cygwin service setup. bz#2922

 * Remove hardcoded service name in cygwin setup. bz#2922

Checksums:
==========

 - SHA1 (openssh-8.0.tar.gz) = 8aaa99091fc7e5a92a4a320e1e5521046b3f95f0
 - SHA256 (openssh-8.0.tar.gz) = 1xvSJk1KYSnOLPYEUzyCVwTEQ7MHOaCO65DzeNuuLdo=

 - SHA1 (openssh-8.0p1.tar.gz) = 756dbb99193f9541c9206a667eaa27b0fa184a4f
 - SHA256 (openssh-8.0p1.tar.gz) = vZQ4eeaUmOgDHra39E0IzcN9WaeraJqgtDcyDDSB/Wg=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available as RELEASE_KEY.asc from
the mirror sites.

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

---

OpenSSH 7.9/7.9p1 (2018-10-19)

OpenSSH 7.9 was released on 2018-10-19. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the

project. More information on donations may be found at:
http://www.openssh.com/donations.html

Potentially-incompatible changes
================================

This release includes a number of changes that may affect existing
configurations:

 * ssh(1), sshd(8): the setting of the new CASignatureAlgorithms
   option (see below) bans the use of DSA keys as certificate
   authorities.

 * sshd(8): the authentication success/failure log message has
   changed format slightly. It now includes the certificate
   fingerprint (previously it included only key ID and CA key
   fingerprint).

Changes since OpenSSH 7.8
=========================

This is primarily a bugfix release.

New Features
----

 * ssh(1), sshd(8): allow most port numbers to be specified using
   service names from getservbyname(3) (typically /etc/services).

 * ssh(1): allow the IdentityAgent configuration directive to accept
   environment variable names. This supports the use of multiple
   agent sockets without needing to use fixed paths.

 * sshd(8): support signalling sessions via the SSH protocol.
   A limited subset of signals is supported and only for login or
   command sessions (i.e. not subsystems) that were not subject to
   a forced command via authorized_keys or sshd_config. bz#1424

 * ssh(1): support "ssh -Q sig" to list supported signature options.
   Also "ssh -Q help" to show the full set of supported queries.

 * ssh(1), sshd(8): add a CASignatureAlgorithms option for the
   client and server configs to allow control over which signature
   formats are allowed for CAs to sign certificates. For example,
   this allows banning CAs that sign certificates using the RSA-SHA1
   signature algorithm.

 * sshd(8), ssh-keygen(1): allow key revocation lists (KRLs) to
   revoke keys specified by SHA256 hash.

 * ssh-keygen(1): allow creation of key revocation lists directly
   from base64-encoded SHA256 fingerprints. This supports revoking
   keys using only the information contained in sshd(8)
   authentication log messages.

Bugfixes
----

 * ssh(1), ssh-keygen(1): avoid spurious "invalid format" errors when
   attempting to load PEM private keys while using an incorrect
   passphrase. bz#2901

 * sshd(8): when a channel closed message is received from a client,
   close the stderr file descriptor at the same time stdout is
   closed. This avoids stuck processes if they were waiting for
   stderr to close and were insensitive to stdin/out closing. bz#2863

 * ssh(1): allow ForwardX11Timeout=0 to disable the untrusted X11
   forwarding timeout and support X11 forwarding indefinitely.
   Previously the behaviour of ForwardX11Timeout=0 was undefined.

 * sshd(8): when compiled with GSSAPI support, cache supported method
   OIDs regardless of whether GSSAPI authentication is enabled in the
   main section of sshd_config. This avoids sandbox violations if
   GSSAPI authentication was later enabled in a Match block. bz#2107

 * sshd(8): do not fail closed when configured with a text key
   revocation list that contains a too-short key. bz#2897

 * ssh(1): treat connections with ProxyJump specified the same as
   ones with a ProxyCommand set with regards to hostname
   canonicalisation (i.e. don't try to canonicalise the hostname
   unless CanonicalizeHostname is set to 'always'). bz#2896

 * ssh(1): fix regression in OpenSSH 7.8 that could prevent public-
   key authentication using certificates hosted in a ssh-agent(1)
   or against sshd(8) from OpenSSH <7.8.

Portability
----

 * All: support building against the openssl-1.1 API (releases 1.1.0g
   and later). The openssl-1.0 API will remain supported at least
   until OpenSSL terminates security patch support for that API version.

 * sshd(8): allow the futex(2) syscall in the Linux seccomp sandbox;
   apparently required by some glibc/OpenSSL combinations.

```
 * sshd(8): handle getgrouplist(3) returning more than
   _SC_NGROUPS_MAX groups. Some platforms consider this limit more
   as a guideline.

Checksums:
==========

 - SHA1 (openssh-7.9.tar.gz) = 7c50a86b8f591decd172ed7f5527abc533098dec
 - SHA256 (openssh-7.9.tar.gz) = nSVigtHGn3+xKXRqpSnp4YOyEPPAb+pCHdWS9Eh/IPY=

 - SHA1 (openssh-7.9p1.tar.gz) = 993aceedea8ecabb1d0dd7293508a361891c4eaa
 - SHA256 (openssh-7.9p1.tar.gz) = a0s7oiU9hO03ccgFByjVl8kc/OiYcTvre2SjBbbxGq0=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available as RELEASE_KEY.asc from
the mirror sites.

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com
```

OpenSSH 7.8/7.8p1 (2018-08-24)

```
OpenSSH 7.8 was released on 2018-08-24. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
http://www.openssh.com/donations.html

Potentially-incompatible changes
================================

This release includes a number of changes that may affect existing
configurations:

 * ssh-keygen(1): write OpenSSH format private keys by default
   instead of using OpenSSL's PEM format. The OpenSSH format,
   supported in OpenSSH releases since 2014 and described in the
   PROTOCOL.key file in the source distribution, offers substantially
   better protection against offline password guessing and supports
   key comments in private keys. If necessary, it is possible to write
   old PEM-style keys by adding "-m PEM" to ssh-keygen's arguments
   when generating or updating a key.

 * sshd(8): remove internal support for S/Key multiple factor
   authentication. S/Key may still be used via PAM or BSD auth.

 * ssh(1): remove vestigal support for running ssh(1) as setuid. This
   used to be required for hostbased authentication and the (long
   gone) rhosts-style authentication, but has not been necessary for
   a long time. Attempting to execute ssh as a setuid binary, or with
   uid != effective uid will now yield a fatal error at runtime.

 * sshd(8): the semantics of PubkeyAcceptedKeyTypes and the similar
   HostbasedAcceptedKeyTypes options have changed. These now specify
   signature algorithms that are accepted for their respective
   authentication mechanism, where previously they specified accepted
   key types. This distinction matters when using the RSA/SHA2
   signature algorithms "rsa-sha2-256", "rsa-sha2-512" and their
   certificate counterparts. Configurations that override these
   options but omit these algorithm names may cause unexpected
   authentication failures (no action is required for configurations
   that accept the default for these options).

 * sshd(8): the precedence of session environment variables has
   changed. ~/.ssh/environment and environment="..." options in
   authorized_keys files can no longer override SSH_* variables set
   implicitly by sshd.

 * ssh(1)/sshd(8): the default IPQoS used by ssh/sshd has changed.
   They will now use DSCP AF21 for interactive traffic and CS1 for
   bulk.  For a detailed rationale, please see the commit message:
   https://cvsweb.openbsd.org/log/src/usr.bin/ssh/readconf.c,v#rev1.284

Changes since OpenSSH 7.7
=========================

This is primarily a bugfix release.

New Features
------------

 * ssh(1)/sshd(8): add new signature algorithms "rsa-sha2-256-cert-
   v01@openssh.com" and "rsa-sha2-512-cert-v01@openssh.com" to
   explicitly force use of RSA/SHA2 signatures in authentication.

 * sshd(8): extend the PermitUserEnvironment option to accept a
   whitelist of environment variable names in addition to global
```

"yes" or "no" settings.

* sshd(8): add a PermitListen directive to sshd_config(5) and a corresponding permitlisten= authorized_keys option that control which listen addresses and port numbers may be used by remote forwarding (ssh -R ...).

* sshd(8): add some countermeasures against timing attacks used for account validation/enumeration. sshd will enforce a minimum time or each failed authentication attempt consisting of a global 5ms minimum plus an additional per-user 0-4ms delay derived from a host secret.

* sshd(8): add a SetEnv directive to allow an administrator to explicitly specify environment variables in sshd_config. Variables set by SetEnv override the default and client-specified environment.

* ssh(1): add a SetEnv directive to request that the server sets an environment variable in the session. Similar to the existing SendEnv option, these variables are set subject to server configuration.

* ssh(1): allow "SendEnv -PATTERN" to clear environment variables previously marked for sending to the server. bz#1285

* ssh(1)/sshd(8): make UID available as a %-expansion everywhere that the username is available currently. bz#2870

* ssh(1): allow setting ProxyJump=none to disable ProxyJump functionality. bz#2869

Bugfixes
----

* sshd(8): avoid observable differences in request parsing that could be used to determine whether a target user is valid.

* all: substantial internal refactoring

* ssh(1)/sshd(8): fix some memory leaks; bz#2366

* ssh(1): fix a pwent clobber (introduced in openssh-7.7) that could occur during key loading, manifesting as crash on some platforms.

* sshd_config(5): clarify documentation for AuthenticationMethods option; bz#2663

* ssh(1): ensure that the public key algorithm sent in a public key SSH_MSG_USERAUTH_REQUEST matches the content of the signature blob. Previously, these could be inconsistent when a legacy or non-OpenSSH ssh-agent returned a RSA/SHA1 signature when asked to make a RSA/SHA2 signature.

* sshd(8): fix failures to read authorized_keys caused by faulty supplemental group caching. bz#2873

* scp(1): apply umask to directories, fixing potential mkdir/chmod race when copying directory trees bz#2839

* ssh-keygen(1): return correct exit code when searching for and hashing known_hosts entries in a single operation; bz#2772

* ssh(1): prefer the ssh binary pointed to via argv[0] to $PATH when re-executing ssh for ProxyJump. bz#2831

* sshd(8): do not ban PTY allocation when a sshd session is restricted because the user password is expired as it breaks password change dialog. (regression in openssh-7.7).

* ssh(1)/sshd(8): fix error reporting from select() failures.

* ssh(1): improve documentation for -w (tunnel) flag, emphasising that -w implicitly sets Tunnel=point-to-point. bz#2365

* ssh-agent(1): implement EMFILE mitigation for ssh-agent. ssh-agent will no longer spin when its file descriptor limit is exceeded. bz#2576

* ssh(1)/sshd(8): disable SSH2_MSG_DEBUG messages for Twisted Conch clients. Twisted Conch versions that lack a version number in their identification strings will mishandle these messages when running on Python 2.x (https://twistedmatrix.com/trac/ticket/9422)
* sftp(1): notify user immediately when underlying ssh process dies expectedly. bz#2719

* ssh(1)/sshd(8): fix tunnel forwarding; regression in 7.7 release. bz#2855

* ssh-agent(1): don't kill ssh-agent's listening socket entirely if it fails to accept(2) a connection. bz#2837

* sshd(8): relax checking of authorized_keys environment="..." options to allow underscores in variable names (regression introduced in 7.7). bz#2851

* ssh(1): add some missing options in the configuration dump output (ssh -G). bz#2835

```
Portability
-----------

 * sshd(8): Expose details of completed authentication to PAM auth
   modules via SSH_AUTH_INFO_0 in the PAM environment. bz#2408

 * Fix compilation problems caused by fights between zlib and OpenSSL
   colliding uses of "free_func"

 * Improve detection of unsupported compiler options. Recently these
   may have manifested as "unsupported -Wl,-z,retpoline" warnings
   during linking.

 * sshd(8): some sandbox support for Linux/s390 bz#2752.

 * regress tests: unbreak key-options.sh test on platforms without
   openpty(3). bz#2856

 * use getrandom(2) for PRNG seeding when built without OpenSSL.

Checksums:
==========

- SHA1 (openssh-7.8.tar.gz) = ed5511cd42b543cd15166a9cbc56705f23b847e7
- SHA256 (openssh-7.8.tar.gz) = TDqIsMEmghsBUNCrSCPyCxChfitntyOLXNC694py1XE

- SHA1 (openssh-7.8p1.tar.gz) = 27e267e370315561de96577fccae563bc2c37a60
- SHA256 (openssh-7.8p1.tar.gz) = GkhLsVFSwYO7JRThEqow3TQTjDz7Ay7uVJCmbFBxRMo

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available as RELEASE_KEY.asc from
the mirror sites.

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com
```

OpenSSH 7.7/7.7p1 (2018-04-02)

```
OpenSSH 7.7 was released on 2018-04-02. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
http://www.openssh.com/donations.html

Potentially-incompatible changes
================================

This release includes a number of changes that may affect existing
configurations:

 * ssh(1)/sshd(8): Drop compatibility support for some very old SSH
   implementations, including ssh.com <=2.* and OpenSSH <= 3.*. These
   versions were all released in or before 2001 and predate the final
   SSH RFCs. The support in question isn't necessary for RFC-compliant
   SSH implementations.

Changes since OpenSSH 7.6
=========================

This is primarily a bugfix release.

New Features
------------

 * All: Add experimental support for PQC XMSS keys (Extended Hash-
   Based Signatures) based on the algorithm described in
   https://tools.ietf.org/html/draft-irtf-cfrg-xmss-hash-based-signatures-12
   The XMSS signature code is experimental and not compiled in by
   default.

 * sshd(8): Add a "rdomain" criteria for the sshd_config Match keyword
   to allow conditional configuration that depends on which routing
   domain a connection was received on (currently supported on OpenBSD
   and Linux).

 * sshd_config(5): Add an optional rdomain qualifier to the
   ListenAddress directive to allow listening on different routing
   domains. This is supported only on OpenBSD and Linux at present.

 * sshd_config(5): Add RDomain directive to allow the authenticated
   session to be placed in an explicit routing domain. This is only
   supported on OpenBSD at present.

 * sshd(8): Add "expiry-time" option for authorized_keys files to
   allow for expiring keys.
```

* ssh(1): Add a BindInterface option to allow binding the outgoing connection to an interface's address (basically a more usable BindAddress)

* ssh(1): Expose device allocated for tun/tap forwarding via a new %T expansion for LocalCommand. This allows LocalCommand to be used to prepare the interface.

* sshd(8): Expose the device allocated for tun/tap forwarding via a new SSH_TUNNEL environment variable. This allows automatic setup of the interface and surrounding network configuration automatically on the server.

* ssh(1)/scp(1)/sftp(1): Add URI support to ssh, sftp and scp, e.g. ssh://user@host or sftp://user@host/path. Additional connection parameters described in draft-ietf-secsh-scp-sftp-ssh-uri-04 are not implemented since the ssh fingerprint format in the draft uses the deprecated MD5 hash with no way to specify the any other algorithm.

* ssh-keygen(1): Allow certificate validity intervals that specify only a start or stop time (instead of both or neither).

* sftp(1): Allow "cd" and "lcd" commands with no explicit path argument. lcd will change to the local user's home directory as usual. cd will change to the starting directory for session (because the protocol offers no way to obtain the remote user's home directory). bz#2760

* sshd(8): When doing a config test with sshd -T, only require the attributes that are actually used in Match criteria rather than (an incomplete list of) all criteria.

Bugfixes
----

* ssh(1)/sshd(8): More strictly check signature types during key exchange against what was negotiated. Prevents downgrade of RSA signatures made with SHA-256/512 to SHA-1.

* sshd(8): Fix support for client that advertise a protocol version of "1.99" (indicating that they are prepared to accept both SSHv1 and SSHv2). This was broken in OpenSSH 7.6 during the removal of SSHv1 support. bz#2810

* ssh(1): Warn when the agent returns a ssh-rsa (SHA1) signature when a rsa-sha2-256/512 signature was requested. This condition is possible when an old or non-OpenSSH agent is in use. bz#2799

* ssh-agent(1): Fix regression introduced in 7.6 that caused ssh-agent to fatally exit if presented an invalid signature request message.

* sshd_config(5): Accept yes/no flag options case-insensitively, as has been the case in ssh_config(5) for a long time. bz#2664

* ssh(1): Improve error reporting for failures during connection. Under some circumstances misleading errors were being shown. bz#2814

* ssh-keyscan(1): Add -D option to allow printing of results directly in SSHFP format. bz#2821

* regress tests: fix PuTTY interop test broken in last release's SSHv1 removal. bz#2823

* ssh(1): Compatibility fix for some servers that erroneously drop the connection when the IUTF8 (RFC8160) option is sent.

* scp(1): Disable RemoteCommand and RequestTTY in the ssh session started by scp (sftp was already doing this.)

* ssh-keygen(1): Refuse to create a certificate with an unusable number of principals.

* ssh-keygen(1): Fatally exit if ssh-keygen is unable to write all the public key during key generation. Previously it would silently ignore errors writing the comment and terminating newline.

* ssh(1): Do not modify hostname arguments that are addresses by automatically forcing them to lower-case. Instead canonicalise them to resolve ambiguities (e.g. ::0001 => ::1) before they are matched against known_hosts. bz#2763

* ssh(1): Don't accept junk after "yes" or "no" responses to hostkey prompts. bz#2803

* sftp(1): Have sftp print a warning about shell cleanliness when decoding the first packet fails, which is usually caused by shells polluting stdout of non-interactive startups. bz#2800

* ssh(1)/sshd(8): Switch timers in packet code from using wall-clock time to monotonic time, allowing the packet layer to better function over a clock step and avoiding possible integer overflows during steps.

* Numerous manual page fixes and improvements.

Portability

```
 * sshd(8): Correctly detect MIPS ABI in use at configure time. Fixes
   sandbox violations on some environments.

 * sshd(8): Remove UNICOS support. The hardware and software are literal
   museum pieces and support in sshd is too intrusive to justify
   maintaining.

 * All: Build and link with "retpoline" flags when available to mitigate
   the "branch target injection" style (variant 2) of the Spectre
   branch-prediction vulnerability.

 * All: Add auto-generated dependency information to Makefile.

 * Numerous fixed to the RPM spec files.

Checksums:
==========

- SHA1 (openssh-7.7.tar.gz) = 24812e05fa233014c847c7775748316e7f8a836c
- SHA256 (openssh-7.7.tar.gz) = T4ua1L/vgAYqwB0muRahvnm5ZUr3PLY9nPljaG8egvo=

- SHA1 (openssh-7.7p1.tar.gz) = 446fe9ed171f289f0d62197dffdbfdaaf21c49f2
- SHA256 (openssh-7.7p1.tar.gz) = 1zvn5oTpnvzQJL4Vowv/y+QbASsvezyQhK7WIXdea48=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available as RELEASE_KEY.asc from
the mirror sites.

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com
```

OpenSSH 7.6/7.6p1 (2017-10-03)

```
OpenSSH 7.6 was released on 2017-10-03. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
http://www.openssh.com/donations.html

Potentially-incompatible changes
================================

This release includes a number of changes that may affect existing
configurations:

 * ssh(1): delete SSH protocol version 1 support, associated
   configuration options and documentation.

 * ssh(1)/sshd(8): remove support for the hmac-ripemd160 MAC.

 * ssh(1)/sshd(8): remove support for the arcfour, blowfish and CAST
   ciphers.

 * Refuse RSA keys <1024 bits in length and improve reporting for keys
   that do not meet this requirement.

 * ssh(1): do not offer CBC ciphers by default.

Changes since OpenSSH 7.5
=========================

This is primarily a bugfix release. It also contains substantial
internal refactoring.

Security
--------

 * sftp-server(8): in read-only mode, sftp-server was incorrectly
   permitting creation of zero-length files. Reported by Michal
   Zalewski.

New Features
------------

 * ssh(1): add RemoteCommand option to specify a command in the ssh
   config file instead of giving it on the client's command line. This
   allows the configuration file to specify the command that will be
   executed on the remote host.

 * sshd(8): add ExposeAuthInfo option that enables writing details of
   the authentication methods used (including public keys where
   applicable) to a file that is exposed via a $SSH_USER_AUTH
   environment variable in the subsequent session.

 * ssh(1): add support for reverse dynamic forwarding. In this mode,
```

ssh will act as a SOCKS4/5 proxy and forward connections
to destinations requested by the remote SOCKS client. This mode
is requested using extended syntax for the -R and RemoteForward
options and, because it is implemented solely at the client,
does not require the server be updated to be supported.

* sshd(8): allow LogLevel directive in sshd_config Match blocks;
bz#2717

* ssh-keygen(1): allow inclusion of arbitrary string or flag
certificate extensions and critical options.

* ssh-keygen(1): allow ssh-keygen to use a key held in ssh-agent as
a CA when signing certificates. bz#2377

* ssh(1)/sshd(8): allow IPQoS=none in ssh/sshd to not set an explicit
ToS/DSCP value and just use the operating system default.

* ssh-add(1): added -q option to make ssh-add quiet on success.

* ssh(1): expand the StrictHostKeyChecking option with two new
settings. The first "accept-new" will automatically accept
hitherto-unseen keys but will refuse connections for changed or
invalid hostkeys. This is a safer subset of the current behaviour
of StrictHostKeyChecking=no. The second setting "off", is a synonym
for the current behaviour of StrictHostKeyChecking=no: accept new
host keys, and continue connection for hosts with incorrect
hostkeys. A future release will change the meaning of
StrictHostKeyChecking=no to the behaviour of "accept-new". bz#2400

* ssh(1): add SyslogFacility option to ssh(1) matching the equivalent
option in sshd(8). bz#2705

Bugfixes
----

* ssh(1): use HostKeyAlias if specified instead of hostname for
matching host certificate principal names; bz#2728

* sftp(1): implement sorting for globbed ls; bz#2649

* ssh(1): add a user@host prefix to client's "Permission denied"
messages, useful in particular when using "stacked" connections
(e.g. ssh -J) where it's not clear which host is denying. bz#2720

* ssh(1): accept unknown EXT_INFO extension values that contain \0
characters. These are legal, but would previously cause fatal
connection errors if received.

* ssh(1)/sshd(8): repair compression statistics printed at
connection exit

* sftp(1): print '?' instead of incorrect link count (that the
protocol doesn't provide) for remote listings. bz#2710

* ssh(1): return failure rather than fatal() for more cases during
session multiplexing negotiations. Causes the session to fall back
to a non-mux connection if they occur. bz#2707

* ssh(1): mention that the server may send debug messages to explain
public key authentication problems under some circumstances; bz#2709

* Translate OpenSSL error codes to better report incorrect passphrase
errors when loading private keys; bz#2699

* sshd(8): adjust compatibility patterns for WinSCP to correctly
identify versions that implement only the legacy DH group exchange
scheme. bz#2748

* ssh(1): print the "Killed by signal 1" message only at LogLevel
verbose so that it is not shown at the default level; prevents it
from appearing during ssh -J and equivalent ProxyCommand configs.
bz#1906, bz#2744

* ssh-keygen(1): when generating all hostkeys (ssh-keygen -A), clobber
existing keys if they exist but are zero length. zero-length keys
could previously be made if ssh-keygen failed or was interrupted part
way through generating them. bz#2561

* ssh(1): fix pledge(2) violation in the escape sequence "~&" used to
place the current session in the background.

* ssh-keyscan(1): avoid double-close() on file descriptors; bz#2734

* sshd(8): avoid reliance on shared use of pointers shared between
monitor and child sshd processes. bz#2704

* sshd_config(8): document available AuthenticationMethods; bz#2453

* ssh(1): avoid truncation in some login prompts; bz#2768

* sshd(8): Fix various compilations failures, inc bz#2767

* ssh(1): make "--" before the hostname terminate argument processing
after the hostname too.

* ssh-keygen(1): switch from aes256-cbc to aes256-ctr for encrypting
new-style private keys. Fixes problems related to private key

handling for no-OpenSSL builds. bz#2754

 * ssh(1): warn and do not attempt to use keys when the public and
   private halves do not match. bz#2737

 * sftp(1): don't print verbose error message when ssh disconnects
   from under sftp. bz#2750

 * sshd(8): fix keepalive scheduling problem: activity on a forwarded
   port from preventing the keepalive from being sent; bz#2756

 * sshd(8): when started without root privileges, don't require the
   privilege separation user or path to exist. Makes running the
   regression tests easier without touching the filesystem.

 * Make integrity.sh regression tests more robust against timeouts.
   bz#2658

 * ssh(1)/sshd(8): correctness fix for channels implementation: accept
   channel IDs greater than 0x7FFFFFFF.

Portability
----

 * sshd(9): drop two more privileges in the Solaris sandbox:
   PRIV_DAX_ACCESS and PRIV_SYS_IB_INFO; bz#2723

 * sshd(8): expose list of completed authentication methods to PAM
   via the SSH_AUTH_INFO_0 PAM environment variable. bz#2408

 * ssh(1)/sshd(8): fix several problems in the tun/tap forwarding code,
   mostly to do with host/network byte order confusion. bz#2735

 * Add --with-cflags-after and --with-ldflags-after configure flags to
   allow setting CFLAGS/LDFLAGS after configure has completed. These
   are useful for setting sanitiser/fuzzing options that may interfere
   with configure's operation.

 * sshd(8): avoid Linux seccomp violations on ppc64le over the
   socketcall syscall.

 * Fix use of ldns when using ldns-config; bz#2697

 * configure: set cache variables when cross-compiling. The cross-
   compiling fallback message was saying it assumed the test passed,
   but it wasn't actually set the cache variables and this would
   cause later tests to fail.

 * Add clang libFuzzer harnesses for public key parsing and signature
   verification.

Checksums:
==========

 - SHA1 (openssh-7.6.tar.gz) = 157fe3989a245c58fcdb34d9fe722a3c4e14c008
 - SHA1 (openssh-7.6p1.tar.gz) = a6984bc2c72192bed015c8b879b35dd9f5350b3b

 - SHA256 (openssh-7.6.tar.gz) = Xu3bdpCcu65vM2FnW7b6IKLgd4Kvf2P3WBTMw+I7Bao=
 - SHA256 (openssh-7.6p1.tar.gz) = oyPK7t3+FFuqoNsW6Y14Sx+8fdQ2pr8fR539XNHSFyM=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available as RELEASE_KEY.asc from
the mirror sites.

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de
Raadt, Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre,
Tim Rice and Ben Lindstrom.

OpenSSH 7.5/7.5p1 (2017-03-20)

OpenSSH 7.5 was released on 2017-03-20. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support. OpenSSH also includes
transitional support for the legacy SSH 1.3 and 1.5 protocols
that may be enabled at compile-time.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
http://www.openssh.com/donations.html

Future deprecation notice
=========================

We plan on retiring more legacy cryptography in future releases,
specifically:

```
   * In the next major release (expected June-August), removing remaining
     support for the SSH v.1 protocol (currently client-only and compile-
     time disabled).

   * In the same release, removing support for Blowfish and RC4 ciphers
     and the RIPE-MD160 HMAC. (These are currently run-time disabled).

   * In the same release, removing the remaining CBC ciphers from being
     offered by default in the client (These have not been offered in
     sshd by default for several years).

   * Refusing all RSA keys smaller than 1024 bits (the current minimum
     is 768 bits)

  This list reflects our current intentions, but please check the final
  release notes for future releases.

  Potentially-incompatible changes
  ================================

  This release includes a number of changes that may affect existing
  configurations:

   * This release deprecates the sshd_config UsePrivilegeSeparation
     option, thereby making privilege separation mandatory. Privilege
     separation has been on by default for almost 15 years and
     sandboxing has been on by default for almost the last five.

   * The format of several log messages emitted by the packet code has
     changed to include additional information about the user and
     their authentication state. Software that monitors ssh/sshd logs
     may need to account for these changes. For example:

     Connection closed by user x 1.1.1.1 port 1234 [preauth]
     Connection closed by authenticating user x 10.1.1.1 port 1234 [preauth]
     Connection closed by invalid user x 1.1.1.1 port 1234 [preauth]

     Affected messages include connection closure, timeout, remote
     disconnection, negotiation failure and some other fatal messages
     generated by the packet code.

   * [Portable OpenSSH only] This version removes support for building
     against OpenSSL versions prior to 1.0.1. OpenSSL stopped supporting
     versions prior to 1.0.1 over 12 months ago (i.e. they no longer
     receive fixes for security bugs).

  Changes since OpenSSH 7.4
  =========================

  This is a bugfix release.

  Security
  --------

   * ssh(1), sshd(8): Fix weakness in CBC padding oracle countermeasures
     that allowed a variant of the attack fixed in OpenSSH 7.3 to proceed.
     Note that the OpenSSH client disables CBC ciphers by default, sshd
     offers them as lowest-preference options and will remove them by
     default entriely in the next release. Reported by Jean Paul
     Degabriele, Kenny Paterson, Martin Albrecht and Torben Hansen of
     Royal Holloway, University of London.

   * sftp-client(1): [portable OpenSSH only] On Cygwin, a client making
     a recursive file transfer could be maniuplated by a hostile server to
     perform a path-traversal attack. creating or modifying files outside
     of the intended target directory. Reported by Jann Horn of Google
     Project Zero.

  New Features
  ------------

   * ssh(1), sshd(8): Support "=-" syntax to easily remove methods from
     algorithm lists, e.g. Ciphers=-*cbc. bz#2671

  Bugfixes
  --------

   * sshd(1): Fix NULL dereference crash when key exchange start
     messages are sent out of sequence.

   * ssh(1), sshd(8): Allow form-feed characters to appear in
     configuration files.

   * sshd(8): Fix regression in OpenSSH 7.4 support for the
     server-sig-algs extension, where SHA2 RSA signature methods were
     not being correctly advertised. bz#2680

   * ssh(1), ssh-keygen(1): Fix a number of case-sensitivity bugs in
     known_hosts processing. bz#2591 bz#2685

   * ssh(1): Allow ssh to use certificates accompanied by a private key
     file but no corresponding plain *.pub public key. bz#2617

   * ssh(1): When updating hostkeys using the UpdateHostKeys option,
     accept RSA keys if HostkeyAlgorithms contains any RSA keytype.
     Previously, ssh could ignore RSA keys when only the ssh-rsa-sha2-*
     methods were enabled in HostkeyAlgorithms and not the old ssh-rsa
     method. bz#2650
```

* ssh(1): Detect and report excessively long configuration file lines. bz#2651

* Merge a number of fixes found by Coverity and reported via Redhat and FreeBSD. Includes fixes for some memory and file descriptor leaks in error paths. bz#2687

* ssh-keyscan(1): Correctly hash hosts with a port number. bz#2692

* ssh(1), sshd(8): When logging long messages to stderr, don't truncate "\r\n" if the length of the message exceeds the buffer. bz#2688

* ssh(1): Fully quote [host]:port in generated ProxyJump/-J command-line; avoid confusion over IPv6 addresses and shells that treat square bracket characters specially.

* ssh-keygen(1): Fix corruption of known_hosts when running "ssh-keygen -H" on a known_hosts containing already-hashed entries.

* Fix various fallout and sharp edges caused by removing SSH protocol 1 support from the server, including the server banner string being incorrectly terminated with only \n (instead of \r\n), confusing error messages from ssh-keyscan bz#2583 and a segfault in sshd if protocol v.1 was enabled for the client and sshd_config contained references to legacy keys bz#2686.

* ssh(1), sshd(8): Free fd_set on connection timeout. bz#2683

* sshd(8): Fix Unix domain socket forwarding for root (regression in OpenSSH 7.4).

* sftp(1): Fix division by zero crash in "df" output when server returns zero total filesystem blocks/inodes.

* ssh(1), ssh-add(1), ssh-keygen(1), sshd(8): Translate OpenSSL errors encountered during key loading to more meaningful error codes. bz#2522 bz#2523

* ssh-keygen(1): Sanitise escape sequences in key comments sent to printf but preserve valid UTF-8 when the locale supports it; bz#2520

* ssh(1), sshd(8): Return reason for port forwarding failures where feasible rather than always "administratively prohibited". bz#2674

* sshd(8): Fix deadlock when AuthorizedKeysCommand or AuthorizedPrincipalsCommand produces a lot of output and a key is matched early. bz#2655

* Regression tests: several reliability fixes. bz#2654 bz#2658 bz#2659

* ssh(1): Fix typo in ~C error message for bad port forward cancellation. bz#2672

* ssh(1): Show a useful error message when included config files can't be opened; bz#2653

* sshd(8): Make sshd set GSSAPIStrictAcceptorCheck=yes as the manual page (previously incorrectly) advertised. bz#2637

* sshd_config(5): Repair accidentally-deleted mention of %k token in AuthorizedKeysCommand; bz#2656

* sshd(8): Remove vestiges of previously removed LOGIN_PROGRAM; bz#2665

* ssh-agent(1): Relax PKCS#11 whitelist to include libexec and common 32-bit compatibility library directories.

* sftp-client(1): Fix non-exploitable integer overflow in SSH2_FXP_NAME response handling.

* ssh-agent(1): Fix regression in 7.4 of deleting PKCS#11-hosted keys. It was not possible to delete them except by specifying their full physical path. bz#2682

Portability
---

* sshd(8): Avoid sandbox errors for Linux S390 systems using an ICA crypto coprocessor.

* sshd(8): Fix non-exploitable weakness in seccomp-bpf sandbox arg inspection.

* ssh(1): Fix X11 forwarding on OSX where X11 was being started by launchd. bz#2341

* ssh-keygen(1), ssh(1), sftp(1): Fix output truncation for various that contain non-printable characters where the codeset in use is ASCII.

* build: Fix builds that attempt to link a kerberised libldns. bz#2603

* build: Fix compilation problems caused by unconditionally defining _XOPEN_SOURCE in wide character detection.

* sshd(8): Fix sandbox violations for clock_gettime VSDO syscall fallback on some Linux/X32 kernels. bz#2142

```
Checksums:
==========

 - SHA1 (openssh-7.5.tar.gz) = 81384df377e38551f7659a4c250383d0bbd25341
 - SHA1 (openssh-7.5p1.tar.gz) = 5e8f185d00afb4f4f89801e9b0f8b9cee9d87ebd

 - SHA256 (openssh-7.5.tar.gz) = Gmk8jOdGdKa7NixUN5J+bTMfeum5Vx8Nv+leAdQNq3U=
 - SHA256 (openssh-7.5p1.tar.gz) = mEbjxfq58FR0ALTSwBeZL5FCIrP9H47ubH3GvF5Z+fA=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available as RELEASE_KEY.asc from
the mirror sites.

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de
Raadt, Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre,
Tim Rice and Ben Lindstrom.
```

OpenSSH 7.4/7.4p1 (2016-12-19)

```
OpenSSH 7.4 was released on 2016-12-19. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support. OpenSSH also includes
transitional support for the legacy SSH 1.3 and 1.5 protocols
that may be enabled at compile-time.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
http://www.openssh.com/donations.html

Future deprecation notice
=========================

We plan on retiring more legacy cryptography in future releases,
specifically:

 * In approximately August 2017, removing remaining support for the
   SSH v.1 protocol (client-only and currently compile-time disabled).

 * In the same release, removing support for Blowfish and RC4 ciphers
   and the RIPE-MD160 HMAC. (These are currently run-time disabled).

 * Refusing all RSA keys smaller than 1024 bits (the current minimum
   is 768 bits)

 * The next release of OpenSSH will remove support for running sshd(8)
   with privilege separation disabled.

 * The next release of portable OpenSSH will remove support for
   OpenSSL version prior to 1.0.1.

This list reflects our current intentions, but please check the final
release notes for future releases.

Potentially-incompatible changes
================================

This release includes a number of changes that may affect existing
configurations:

 * This release removes server support for the SSH v.1 protocol.

 * ssh(1): Remove 3des-cbc from the client's default proposal. 64-bit
   block ciphers are not safe in 2016 and we don't want to wait until
   attacks like SWEET32 are extended to SSH. As 3des-cbc was the
   only mandatory cipher in the SSH RFCs, this may cause problems
   connecting to older devices using the default configuration,
   but it's highly likely that such devices already need explicit
   configuration for key exchange and hostkey algorithms already
   anyway.

 * sshd(8): Remove support for pre-authentication compression.
   Doing compression early in the protocol probably seemed reasonable
   in the 1990s, but today it's clearly a bad idea in terms of both
   cryptography (cf. multiple compression oracle attacks in TLS) and
   attack surface. Pre-auth compression support has been disabled by
   default for >10 years. Support remains in the client.

 * ssh-agent will refuse to load PKCS#11 modules outside a whitelist
   of trusted paths by default. The path whitelist may be specified
   at run-time.

 * sshd(8): When a forced-command appears in both a certificate and
   an authorized keys/principals command= restriction, sshd will now
   refuse to accept the certificate unless they are identical.
```

The previous (documented) behaviour of having the certificate forced-command override the other could be a bit confusing and error-prone.

* sshd(8): Remove the UseLogin configuration directive and support for having /bin/login manage login sessions.

Changes since OpenSSH 7.3
=========================

This is primarily a bugfix release.

Security
----

* ssh-agent(1): Will now refuse to load PKCS#11 modules from paths outside a trusted whitelist (run-time configurable). Requests to load modules could be passed via agent forwarding and an attacker could attempt to load a hostile PKCS#11 module across the forwarded agent channel: PKCS#11 modules are shared libraries, so this would result in code execution on the system running the ssh-agent if the attacker has control of the forwarded agent-socket (on the host running the sshd server) and the ability to write to the filesystem of the host running ssh-agent (usually the host running the ssh client). Reported by Jann Horn of Project Zero.

* sshd(8): When privilege separation is disabled, forwarded Unix-domain sockets would be created by sshd(8) with the privileges of 'root' instead of the authenticated user. This release refuses Unix-domain socket forwarding when privilege separation is disabled (Privilege separation has been enabled by default for 14 years). Reported by Jann Horn of Project Zero.

* sshd(8): Avoid theoretical leak of host private key material to privilege-separated child processes via realloc() when reading keys. No such leak was observed in practice for normal-sized keys, nor does a leak to the child processes directly expose key material to unprivileged users. Reported by Jann Horn of Project Zero.

* sshd(8): The shared memory manager used by pre-authentication compression support had a bounds checks that could be elided by some optimising compilers. Additionally, this memory manager was incorrectly accessible when pre-authentication compression was disabled. This could potentially allow attacks against the privileged monitor process from the sandboxed privilege-separation process (a compromise of the latter would be required first). This release removes support for pre-authentication compression from sshd(8). Reported by Guido Vranken using the Stack unstable optimisation identification tool (http://css.csail.mit.edu/stack/)
* sshd(8): Fix denial-of-service condition where an attacker who sends multiple KEXINIT messages may consume up to 128MB per connection. Reported by Shi Lei of Gear Team, Qihoo 360.

* sshd(8): Validate address ranges for AllowUser and DenyUsers directives at configuration load time and refuse to accept invalid ones. It was previously possible to specify invalid CIDR address ranges (e.g. user@127.1.2.3/55) and these would always match, possibly resulting in granting access where it was not intended. Reported by Laurence Parry.

New Features
----

* ssh(1): Add a proxy multiplexing mode to ssh(1) inspired by the version in PuTTY by Simon Tatham. This allows a multiplexing client to communicate with the master process using a subset of the SSH packet and channels protocol over a Unix-domain socket, with the main process acting as a proxy that translates channel IDs, etc. This allows multiplexing mode to run on systems that lack file- descriptor passing (used by current multiplexing code) and potentially, in conjunction with Unix-domain socket forwarding, with the client and multiplexing master process on different machines. Multiplexing proxy mode may be invoked using "ssh -O proxy ..."

* sshd(8): Add a sshd_config DisableForwarding option that disables X11, agent, TCP, tunnel and Unix domain socket forwarding, as well as anything else we might implement in the future. Like the 'restrict' authorized_keys flag, this is intended to be a simple and future-proof way of restricting an account.

* sshd(8), ssh(1): Support the "curve25519-sha256" key exchange method. This is identical to the currently-supported method named "curve25519-sha256@libssh.org".

* sshd(8): Improve handling of SIGHUP by checking to see if sshd is already daemonised at startup and skipping the call to daemon(3) if it is. This ensures that a SIGHUP restart of sshd(8) will retain the same process-ID as the initial execution. sshd(8) will also now unlink the PidFile prior to SIGHUP restart and re-create it after a successful restart, rather than leaving a stale file in the case of a configuration error. bz#2641

* sshd(8): Allow ClientAliveInterval and ClientAliveCountMax directives to appear in sshd_config Match blocks.

* sshd(8): Add %-escapes to AuthorizedPrincipalsCommand to match those supported by AuthorizedKeysCommand (key, key type,

```
    fingerprint, etc.) and a few more to provide access to the
    contents of the certificate being offered.

  * Added regression tests for string matching, address matching and
    string sanitisation functions.

  * Improved the key exchange fuzzer harness.

Bugfixes
--------

  * ssh(1): Allow IdentityFile to successfully load and use
    certificates that have no corresponding bare public key. bz#2617
    certificate id_rsa-cert.pub (and no id_rsa.pub).

  * ssh(1): Fix public key authentication when multiple
    authentication is in use and publickey is not just the first
    method attempted. bz#2642

  * regress: Allow the PuTTY interop tests to run unattended. bz#2639

  * ssh-agent(1), ssh(1): improve reporting when attempting to load
    keys from PKCS#11 tokens with fewer useless log messages and more
    detail in debug messages. bz#2610

  * ssh(1): When tearing down ControlMaster connections, don't
    pollute stderr when LogLevel=quiet.

  * sftp(1): On ^Z wait for underlying ssh(1) to suspend before
    suspending sftp(1) to ensure that ssh(1) restores the terminal mode
    correctly if suspended during a password prompt.

  * ssh(1): Avoid busy-wait when ssh(1) is suspended during a password
    prompt.

  * ssh(1), sshd(8): Correctly report errors during sending of ext-
    info messages.

  * sshd(8): fix NULL-deref crash if sshd(8) received an out-of-
    sequence NEWKEYS message.

  * sshd(8): Correct list of supported signature algorithms sent in
    the server-sig-algs extension. bz#2547

  * sshd(8): Fix sending ext_info message if privsep is disabled.

  * sshd(8): more strictly enforce the expected ordering of privilege
    separation monitor calls used for authentication and allow them
    only when their respective authentication methods are enabled
    in the configuration

  * sshd(8): Fix uninitialised optlen in getsockopt() call; harmless
    on Unix/BSD but potentially crashy on Cygwin.

  * Fix false positive reports caused by explicit_bzero(3) not being
    recognised as a memory initialiser when compiled with
    -fsanitize-memory.

  * sshd_config(5): Use 2001:db8::/32, the official IPv6 subnet for
    configuration examples.

Portability
-----------

  * On environments configured with Turkish locales, fall back to the
    C/POSIX locale to avoid errors in configuration parsing caused by
    that locale's unique handling of the letters 'i' and 'I'. bz#2643

  * sftp-server(8), ssh-agent(1): Deny ptrace on OS X using
    ptrace(PT_DENY_ATTACH, ..)

  * ssh(1), sshd(8): Unbreak AES-CTR ciphers on old (~0.9.8) OpenSSL.

  * Fix compilation for libcrypto compiled without RIPEMD160 support.

  * contrib: Add a gnome-ssh-askpass3 with GTK+3 support. bz#2640

  * sshd(8): Improve PRNG reseeding across privilege separation and
    force libcrypto to obtain a high-quality seed before chroot or
    sandboxing.

  * All: Explicitly test for broken strnvis. NetBSD added an strnvis
    and unfortunately made it incompatible with the existing one in
    OpenBSD and Linux's libbsd (the former having existed for over ten
    years). Try to detect this mess, and assume the only safe option
    if we're cross compiling.

Checksums:
==========

  - SHA1 (openssh-7.4.tar.gz) = 1e2073f95d5ead8f2814b4b6c0700bcd533c410f
  - SHA1 (openssh-7.4p1.tar.gz) = 2330bbf82ed08cf3ac70e0acf00186ef3eeb97e0

  - SHA256 (openssh-7.4.tar.gz) = +GEXh7Xr2J87cq1uA97hF9e+3lfOQ2LKxXGdmFXREf0
  - SHA256 (openssh-7.4p1.tar.gz) = Gx/EoU4gJCkxgZJO0khy5vLgYpPz6JJqN2uK7EgfGdE=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
```

key used to sign the releases is available as RELEASE_KEY.asc from
the mirror sites.

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de
Raadt, Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre,
Tim Rice and Ben Lindstrom.

OpenSSH 7.3/7.3p1 (2016-08-01)

OpenSSH 7.3 was released on 2016-08-01. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support. OpenSSH also includes
transitional support for the legacy SSH 1.3 and 1.5 protocols
that may be enabled at compile-time.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
http://www.openssh.com/donations.html

Future deprecation notice
=========================

We plan on retiring more legacy cryptography in a near-future
release, specifically:

 * Refusing all RSA keys smaller than 1024 bits (the current minimum
   is 768 bits)
 * Removing server-side support for the SSH v.1 protocol (currently
   compile-time disabled).
 * In approximately 1 year, removing all support for the SSH v.1
   protocol (currently compile-time disabled).

This list reflects our current intentions, but please check the final
release notes for future releases.

Changes since OpenSSH 7.2
=========================

This is primarily a bugfix release.

Security
----

 * sshd(8): Mitigate a potential denial-of-service attack against
   the system's crypt(3) function via sshd(8). An attacker could
   send very long passwords that would cause excessive CPU use in
   crypt(3). sshd(8) now refuses to accept password authentication
   requests of length greater than 1024 characters. Independently
   reported by Tomas Kuthan (Oracle), Andres Rojas and Javier Nieto.

 * sshd(8): Mitigate timing differences in password authentication
   that could be used to discern valid from invalid account names
   when long passwords were sent and particular password hashing
   algorithms are in use on the server. CVE-2016-6210, reported by
   EddieEzra.Harari at verint.com

 * ssh(1), sshd(8): Fix observable timing weakness in the CBC padding
   oracle countermeasures. Reported by Jean Paul Degabriele, Kenny
   Paterson, Torben Hansen and Martin Albrecht. Note that CBC ciphers
   are disabled by default and only included for legacy compatibility.

 * ssh(1), sshd(8): Improve operation ordering of MAC verification for
   Encrypt-then-MAC (EtM) mode transport MAC algorithms to verify the
   MAC before decrypting any ciphertext. This removes the possibility
   of timing differences leaking facts about the plaintext, though no
   such leakage has been observed. Reported by Jean Paul Degabriele,
   Kenny Paterson, Torben Hansen and Martin Albrecht.

 * sshd(8): (portable only) Ignore PAM environment vars when
   UseLogin=yes. If PAM is configured to read user-specified
   environment variables and UseLogin=yes in sshd_config, then a
   hostile local user may attack /bin/login via LD_PRELOAD or
   similar environment variables set via PAM. CVE-2015-8325,
   found by Shayan Sadigh.

New Features
----

 * ssh(1): Add a ProxyJump option and corresponding -J command-line
   flag to allow simplified indirection through a one or more SSH
   bastions or "jump hosts".

 * ssh(1): Add an IdentityAgent option to allow specifying specific
   agent sockets instead of accepting one from the environment.

 * ssh(1): Allow ExitOnForwardFailure and ClearAllForwardings to be

```
    optionally overridden when using ssh -W. bz#2577

 * ssh(1), sshd(8): Implement support for the IUTF8 terminal mode as
   per draft-sgtatham-secsh-iutf8-00.

 * ssh(1), sshd(8): Add support for additional fixed Diffie-Hellman
   2K, 4K and 8K groups from draft-ietf-curdle-ssh-kex-sha2-03.

 * ssh-keygen(1), ssh(1), sshd(8): support SHA256 and SHA512 RSA
   signatures in certificates;

 * ssh(1): Add an Include directive for ssh_config(5) files.

 * ssh(1): Permit UTF-8 characters in pre-authentication banners sent
   from the server. bz#2058

Bugfixes
--------

 * ssh(1), sshd(8): Reduce the syslog level of some relatively common
   protocol events from LOG_CRIT. bz#2585

 * sshd(8): Refuse AuthenticationMethods="" in configurations and
   accept AuthenticationMethods=any for the default behaviour of not
   requiring multiple authentication. bz#2398

 * sshd(8): Remove obsolete and misleading "POSSIBLE BREAK-IN
   ATTEMPT!" message when forward and reverse DNS don't match. bz#2585

 * ssh(1): Close ControlPersist background process stderr except
   in debug mode or when logging to syslog. bz#1988

 * misc: Make PROTOCOL description for direct-streamlocal@openssh.com
   channel open messages match deployed code. bz#2529

 * ssh(1): Deduplicate LocalForward and RemoteForward entries to fix
   failures when both ExitOnForwardFailure and hostname
   canonicalisation are enabled. bz#2562

 * sshd(8): Remove fallback from moduli to obsolete "primes" file
   that was deprecated in 2001. bz#2559.

 * sshd_config(5): Correct description of UseDNS: it affects ssh
   hostname processing for authorized_keys, not known_hosts; bz#2554

 * ssh(1): Fix authentication using lone certificate keys in an agent
   without corresponding private keys on the filesystem. bz#2550

 * sshd(8): Send ClientAliveInterval pings when a time-based
   RekeyLimit is set; previously keepalive packets were not being
   sent. bz#2252

Portability
-----------

 * ssh(1), sshd(8): Fix compilation by automatically disabling ciphers
   not supported by OpenSSL. bz#2466

 * misc: Fix compilation failures on some versions of AIX's compiler
   related to the definition of the VA_COPY macro. bz#2589

 * sshd(8): Whitelist more architectures to enable the seccomp-bpf
   sandbox. bz#2590

 * ssh-agent(1), sftp-server(8): Disable process tracing on Solaris
   using setpflags(__PROC_PROTECT, ...). bz#2584

 * sshd(8): On Solaris, don't call Solaris setproject() with
   UsePAM=yes it's PAM's responsibility. bz#2425

Checksums:
==========

 - SHA1 (openssh-7.3.tar.gz) = b1641e5265d9ec68a9a19decc3a7edd1203cbd33
 - SHA256 (openssh-7.3.tar.gz) = vS0X35qrX9OOPBkyDMYhOje/DBwHBVEV7nv5rkzw4vM=

 - SHA1 (openssh-7.3p1.tar.gz) = bfade84283fcba885e2084343ab19a08c7d123a5
 - SHA256 (openssh-7.3p1.tar.gz) = P/uYmm3KppWUw7VQ1IVaWi4XGMzd5/XjY4e0JCIPvsw=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available as RELEASE_KEY.asc from
the mirror sites.

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de
Raadt, Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre,
Tim Rice and Ben Lindstrom.
```

OpenSSH 7.2p2 (2016-03-10)

Portable OpenSSH 7.2p2 was released on 2016-03-10. It will be available
from the mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support. OpenSSH also includes
transitional support for the legacy SSH 1.3 and 1.5 protocols that
may be enabled at compile-time.

Once again, we would like to thank the OpenSSH community for
their continued support of the project, especially those who
contributed code or patches, reported bugs, tested snapshots or
donated to the project. More information on donations may be found
at: http://www.openssh.com/donations.html

Changes since OpenSSH 7.2p1
===========================

This release fixes a security bug:

 * sshd(8): sanitise X11 authentication credentials to avoid xauth
   command injection when X11Forwarding is enabled.

   Full details of the vulnerability are available at:
   http://www.openssh.com/txt/x11fwd.adv

Checksums:
==========

 - SHA1 (openssh-7.2p2.tar.gz) = 70e35d7d6386fe08abbd823b3a12a3ca44ac6d38
 - SHA256 (openssh-7.2p2.tar.gz) = pyeB0aBDh2oiT/GwAy2qQJTYdWWmhSh1nBwsq1SCVIw=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available as RELEASE_KEY.asc from
the mirror sites.

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de
Raadt, Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre,
Tim Rice and Ben Lindstrom.

OpenSSH 7.2/7.2p1 (2016-02-29)

OpenSSH 7.2 was released on 2016-02-29. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support. OpenSSH also includes
transitional support for the legacy SSH 1.3 and 1.5 protocols
that may be enabled at compile-time.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
http://www.openssh.com/donations.html

Future deprecation notice
=========================

We plan on retiring more legacy cryptography in a near-future
release, specifically:

 * Refusing all RSA keys smaller than 1024 bits (the current minimum
   is 768 bits)

This list reflects our current intentions, but please check the final
release notes for future releases.

Potentially-incompatible changes
================================

This release disables a number of legacy cryptographic algorithms
by default in ssh:

 * Several ciphers blowfish-cbc, cast128-cbc, all arcfour variants
   and the rijndael-cbc aliases for AES.

 * MD5-based and truncated HMAC algorithms.

These algorithms are already disabled by default in sshd.

Changes since OpenSSH 7.1p2
===========================

This is primarily a bugfix release.

Security
----

 * ssh(1), sshd(8): remove unfinished and unused roaming code (was
   already forcibly disabled in OpenSSH 7.1p2).

 * ssh(1): eliminate fallback from untrusted X11 forwarding to
   trusted forwarding when the X server disables the SECURITY
   extension.

 * ssh(1), sshd(8): increase the minimum modulus size supported for
   diffie-hellman-group-exchange to 2048 bits.

 * sshd(8): pre-auth sandboxing is now enabled by default (previous
   releases enabled it for new installations via sshd_config).

New Features
----

 * all: add support for RSA signatures using SHA-256/512 hash
   algorithms based on draft-rsa-dsa-sha2-256-03.txt and
   draft-ssh-ext-info-04.txt.

 * ssh(1): Add an AddKeysToAgent client option which can be set to
   'yes', 'no', 'ask', or 'confirm', and defaults to 'no'. When
   enabled, a private key that is used during authentication will be
   added to ssh-agent if it is running (with confirmation enabled if
   set to 'confirm').

 * sshd(8): add a new authorized_keys option "restrict" that includes
   all current and future key restrictions (no-*-forwarding, etc.).
   Also add permissive versions of the existing restrictions, e.g.
   "no-pty" -> "pty". This simplifies the task of setting up
   restricted keys and ensures they are maximally-restricted,
   regardless of any permissions we might implement in the future.

 * ssh(1): add ssh_config CertificateFile option to explicitly list
   certificates. bz#2436

 * ssh-keygen(1): allow ssh-keygen to change the key comment for all
   supported formats.

 * ssh-keygen(1): allow fingerprinting from standard input, e.g.
   "ssh-keygen -lf -"

 * ssh-keygen(1): allow fingerprinting multiple public keys in a
   file, e.g. "ssh-keygen -lf ~/.ssh/authorized_keys" bz#1319

 * sshd(8): support "none" as an argument for sshd_config
   Foreground and ChrootDirectory. Useful inside Match blocks to
   override a global default. bz#2486

 * ssh-keygen(1): support multiple certificates (one per line) and
   reading from standard input (using "-f -") for "ssh-keygen -L"

 * ssh-keyscan(1): add "ssh-keyscan -c ..." flag to allow fetching
   certificates instead of plain keys.

 * ssh(1): better handle anchored FQDNs (e.g. 'cvs.openbsd.org.') in
   hostname canonicalisation - treat them as already canonical and
   remove the trailing '.' before matching ssh_config.

Bugfixes
----

 * sftp(1): existing destination directories should not terminate
   recursive uploads (regression in openssh 6.8) bz#2528

 * ssh(1), sshd(8): correctly send back SSH2_MSG_UNIMPLEMENTED
   replies to unexpected messages during key exchange. bz#2949

 * ssh(1): refuse attempts to set ConnectionAttempts=0, which does
   not make sense and would cause ssh to print an uninitialised stack
   variable. bz#2500

 * ssh(1): fix errors when attempting to connect to scoped IPv6
   addresses with hostname canonicalisation enabled.

 * sshd_config(5): list a couple more options usable in Match blocks.
   bz#2489

 * sshd(8): fix "PubkeyAcceptedKeyTypes +..." inside a Match block.

 * ssh(1): expand tilde characters in filenames passed to -i options
   before checking whether or not the identity file exists. Avoids
   confusion for cases where shell doesn't expand (e.g. "-i ~/file"
   vs. "-i~/file"). bz#2481

 * ssh(1): do not prepend "exec" to the shell command run by "Match
   exec" in a config file, which could cause some commands to fail
   in certain environments. bz#2471

 * ssh-keyscan(1): fix output for multiple hosts/addrs on one line
   when host hashing or a non standard port is in use bz#2479

 * sshd(8): skip "Could not chdir to home directory" message when
   ChrootDirectory is active. bz#2485

 * ssh(1): include PubkeyAcceptedKeyTypes in ssh -G config dump.

 * sshd(8): avoid changing TunnelForwarding device flags if they are
   already what is needed; makes it possible to use tun/tap
   networking as non-root user if device permissions and interface

flags are pre-established

* ssh(1), sshd(8): RekeyLimits could be exceeded by one packet. bz#2521

* ssh(1): fix multiplexing master failure to notice client exit.

* ssh(1), ssh-agent(1): avoid fatal() for PKCS11 tokens that present empty key IDs. bz#1773

* sshd(8): avoid printf of NULL argument. bz#2535

* ssh(1), sshd(8): allow RekeyLimits larger than 4GB. bz#2521

* ssh-keygen(1): sshd(8): fix several bugs in (unused) KRL signature support.

* ssh(1), sshd(8): fix connections with peers that use the key exchange guess feature of the protocol. bz#2515

* sshd(8): include remote port number in log messages. bz#2503

* ssh(1): don't try to load SSHv1 private key when compiled without SSHv1 support. bz#2505

* ssh-agent(1), ssh(1): fix incorrect error messages during key loading and signing errors. bz#2507

* ssh-keygen(1): don't leave empty temporary files when performing known_hosts file edits when known_hosts doesn't exist.

* sshd(8): correct packet format for tcpip-forward replies for requests that don't allocate a port bz#2509

* ssh(1), sshd(8): fix possible hang on closed output. bz#2469

* ssh(1): expand %i in ControlPath to UID. bz#2449

* ssh(1), sshd(8): fix return type of openssh_RSA_verify. bz#2460

* ssh(1), sshd(8): fix some option parsing memory leaks. bz#2182

* ssh(1): add a some debug output before DNS resolution; it's a place where ssh could previously silently stall in cases of unresponsive DNS servers. bz#2433

* ssh(1): remove spurious newline in visual hostkey. bz#2686

* ssh(1): fix printing (ssh -G ...) of HostKeyAlgorithms=+...

* ssh(1): fix expansion of HostkeyAlgorithms=+...

Documentation
---

* ssh_config(5), sshd_config(5): update default algorithm lists to match current reality. bz#2527

* ssh(1): mention -Q key-plain and -Q key-cert query options. bz#2455

* sshd_config(8): more clearly describe what AuthorizedKeysFile=none does.

* ssh_config(5): better document ExitOnForwardFailure. bz#2444

* sshd(5): mention internal DH-GEX fallback groups in manual. bz#2302

* sshd_config(5): better description for MaxSessions option. bz#2531

Portability
---

* ssh(1), sftp-server(8), ssh-agent(1), sshd(8): Support Illumos/ Solaris fine-grained privileges. Including a pre-auth privsep sandbox and several pledge() emulations. bz#2511

* Renovate redhat/openssh.spec, removing deprecated options and syntax.

* configure: allow --without-ssl-engine with --without-openssl

* sshd(8): fix multiple authentication using S/Key. bz#2502

* sshd(8): read back from libcrypto RAND_* before dropping privileges. Avoids sandboxing violations with BoringSSL.

* Fix name collision with system-provided glob(3) functions. bz#2463

* Adapt Makefile to use ssh-keygen -A when generating host keys. bz#2459

* configure: correct default value for --with-ssh1 bz#2457

* configure: better detection of _res symbol bz#2259

```
 * support getrandom() syscall on Linux

Checksums:
==========

 - SHA1 (openssh-7.2.tar.gz) = 9567d00fffe655010c087aeb80c830cecbbecca6
 - SHA256 (openssh-7.2.tar.gz) = 99GsHA8NwSGuEJhMc7hAOQ510y1xfGx27uJqyw73sCI=

 - SHA1 (openssh-7.2p1.tar.gz) = d30a6fd472199ab5838a7668c0c5fd885fb8d371
 - SHA256 (openssh-7.2p1.tar.gz) = lzzDey81l+TPWZsJ5gTnnA/l2bb1laJOke0GYoYLSsM=

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available as RELEASE_KEY.asc from
the mirror sites.

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de
Raadt, Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre,
Tim Rice and Ben Lindstrom.
```

OpenSSH 7.1p2 (2016-01-14)

```
OpenSSH 7.1p2 was released on 2016-01-14. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support. OpenSSH also includes
transitional support for the legacy SSH 1.3 and 1.5 protocols
that may be enabled at compile-time.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
http://www.openssh.com/donations.html

Changes since OpenSSH 7.1p1
===========================

 * SECURITY: ssh(1): The OpenSSH client code between 5.4 and 7.1
   contains experimental support for resuming SSH-connections (roaming).

   The matching server code has never been shipped, but the client
   code was enabled by default and could be tricked by a malicious
   server into leaking client memory to the server, including private
   client user keys.

   The authentication of the server host key prevents exploitation
   by a man-in-the-middle, so this information leak is restricted
   to connections to malicious or compromised servers.

   MITIGATION: For OpenSSH >= 5.4 the vulnerable code in the client
   can be completely disabled by adding 'UseRoaming no' to the gobal
   ssh_config(5) file, or to user configuration in ~/.ssh/config,
   or by passing -oUseRoaming=no on the command line.

   PATCH: See below for a patch to disable this feature (Disabling
   Roaming in the Source Code).

   This problem was reported by the Qualys Security Advisory team.

 * SECURITY: Fix an out of-bound read access in the packet handling
   code. Reported by Ben Hawkes.

 * PROTOCOL: Correctly interpret the 'first_kex_follows' option during
   the intial key exchange. Reported by Matt Johnston.

 * Further use of explicit_bzero has been added in various buffer
   handling code paths to guard against compilers aggressively
   doing dead-store removal.


Checksums:
==========

 - SHA1 (openssh-7.1p2.tar.gz) = 9202f5a2a50c8a55ecfb830609df1e1fde97f758
 - SHA256 (openssh-7.1p2.tar.gz) = dd75f024dcf21e06a0d6421d582690bf987a1f6323e32ad6619392f3bfde6bbd

Please note that the SHA256 signatures are base64 encoded and not
hexadecimal (which is the default for most checksum tools). The PGP
key used to sign the releases is available as RELEASE_KEY.asc from
the mirror sites.

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com
```

```
OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.

Disabling Roaming in the Source Code:
=====================================

--- readconf.c  30 Jul 2015 00:01:34 -0000      1.239
+++ readconf.c  13 Jan 2016 23:17:23 -0000
@@ -1648,7 +1648,7 @@ initialize_options(Options * options)
        options->tun_remote = -1;
        options->local_command = NULL;
        options->permit_local_command = -1;
-       options->use_roaming = -1;
+       options->use_roaming = 0;
        options->visual_host_key = -1;
        options->ip_qos_interactive = -1;
        options->ip_qos_bulk = -1;
@@ -1819,8 +1819,7 @@ fill_default_options(Options * options)
                options->tun_remote = SSH_TUNID_ANY;
        if (options->permit_local_command == -1)
                options->permit_local_command = 0;
-       if (options->use_roaming == -1)
-               options->use_roaming = 1;
+       options->use_roaming = 0;
        if (options->visual_host_key == -1)
                options->visual_host_key = 0;
        if (options->ip_qos_interactive == -1)
--- ssh.c       30 Jul 2015 00:01:34 -0000      1.420
+++ ssh.c       13 Jan 2016 23:17:23 -0000
@@ -1882,9 +1882,6 @@ ssh_session2(void)
                        fork_postauth();
        }

-       if (options.use_roaming)
-               request_roaming();
-
        return client_loop(tty_flag, tty_flag ?
            options.escape_char : SSH_ESCAPECHAR_NONE, id);
 }
```

OpenSSH 7.1/7.1p1 (2015-08-21)

```
OpenSSH 7.1 was released on 2015-08-21. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and
includes sftp client and server support. OpenSSH also includes
transitional support for the legacy SSH 1.3 and 1.5 protocols
that may be enabled at compile-time.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
http://www.openssh.com/donations.html

Future deprecation notice
=========================

We plan on retiring more legacy cryptography in the next release
including:

 * Refusing all RSA keys smaller than 1024 bits (the current minimum
   is 768 bits)

 * Several ciphers will be disabled by default: blowfish-cbc,
   cast128-cbc, all arcfour variants and the rijndael-cbc aliases
   for AES.

 * MD5-based HMAC algorithms will be disabled by default.

This list reflects our current intentions, but please check the final
release notes for OpenSSH 7.2 when it is released.

Changes since OpenSSH 7.0
=========================

This is a bugfix release.

Security
--------

 * sshd(8): OpenSSH 7.0 contained a logic error in PermitRootLogin=
   prohibit-password/without-password that could, depending on
   compile-time configuration, permit password authentication to
   root while preventing other forms of authentication. This problem
   was reported by Mantas Mikulenas.

Bugfixes
--------

 * ssh(1), sshd(8): add compatibility workarounds for FuTTY

 * ssh(1), sshd(8): refine compatibility workarounds for WinSCP

 * Fix a number of memory faults (double-free, free of uninitialised
```

memory, etc) in ssh(1) and ssh-keygen(1). Reported by Mateusz Kocielski.

Checksums:
==========

- SHA1 (openssh-7.1.tar.gz) = 06c1db39f33831fe004726e013b2cf84f1889042
- SHA256 (openssh-7.1.tar.gz) = H7U1se9EoBmhkKi2i7lqpMX9QHdDTsgpu7kd5VZUGSY=

- SHA1 (openssh-7.1p1.tar.gz) = ed22af19f962262c493fcc6ed8c8826b2761d9b6
- SHA256 (openssh-7.1p1.tar.gz) = /AptLR0GPVxm3/2VJJPQzaJWytIE9oHeD4TvhbKthCg=

Please note that the SHA256 signatures are base64 encoded and not hexadecimal (which is the default for most checksum tools). The PGP key used to sign the releases is available as RELEASE_KEY.asc from the mirror sites.

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt, Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and Ben Lindstrom.

OpenSSH 7.0/7.0p1 (2015-08-11)

OpenSSH 7.0 was released on 2015-08-11. It is available from the mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and includes sftp client and server support. OpenSSH also includes transitional support for the legacy SSH 1.3 and 1.5 protocols that may be enabled at compile-time.

Once again, we would like to thank the OpenSSH community for their continued support of the project, especially those who contributed code or patches, reported bugs, tested snapshots or donated to the project. More information on donations may be found at:
http://www.openssh.com/donations.html

Future deprecation notice
=========================

We plan on retiring more legacy cryptography in the next release including:

 * Refusing all RSA keys smaller than 1024 bits (the current minimum is 768 bits)

 * Several ciphers will be disabled by default: blowfish-cbc, cast128-cbc, all arcfour variants and the rijndael-cbc aliases for AES.

 * MD5-based HMAC algorithms will be disabled by default.

This list reflects our current intentions, but please check the final release notes for OpenSSH 7.1 when it is released.

Changes since OpenSSH 6.9
=========================

This focus of this release is primarily to deprecate weak, legacy and/or unsafe cryptography.

Security
----

 * sshd(8): OpenSSH 6.8 and 6.9 incorrectly set TTYs to be world-writable. Local attackers may be able to write arbitrary messages to logged-in users, including terminal escape sequences. Reported by Nikolay Edigaryev.

 * sshd(8): Portable OpenSSH only: Fixed a privilege separation weakness related to PAM support. Attackers who could successfully compromise the pre-authentication process for remote code execution and who had valid credentials on the host could impersonate other users. Reported by Moritz Jodeit.

 * sshd(8): Portable OpenSSH only: Fixed a use-after-free bug related to PAM support that was reachable by attackers who could compromise the pre-authentication process for remote code execution. Also reported by Moritz Jodeit.

 * sshd(8): fix circumvention of MaxAuthTries using keyboard-interactive authentication. By specifying a long, repeating keyboard-interactive "devices" string, an attacker could request the same authentication method be tried thousands of times in a single pass. The LoginGraceTime timeout in sshd(8) and any authentication failure delays implemented by the authentication mechanism itself were still applied. Found by Kingcope.

Potentially-incompatible Changes

```
 * Support for the legacy SSH version 1 protocol is disabled by
   default at compile time.

 * Support for the 1024-bit diffie-hellman-group1-sha1 key exchange
   is disabled by default at run-time. It may be re-enabled using
   the instructions at http://www.openssh.com/legacy.html

 * Support for ssh-dss, ssh-dss-cert-* host and user keys is disabled
   by default at run-time. These may be re-enabled using the
   instructions at http://www.openssh.com/legacy.html

 * Support for the legacy v00 cert format has been removed.

 * The default for the sshd_config(5) PermitRootLogin option has
   changed from "yes" to "prohibit-password".

 * PermitRootLogin=without-password/prohibit-password now bans all
   interactive authentication methods, allowing only public-key,
   hostbased and GSSAPI authentication (previously it permitted
   keyboard-interactive and password-less authentication if those
   were enabled).

New Features
------------

 * ssh_config(5): add PubkeyAcceptedKeyTypes option to control which
   public key types are available for user authentication.

 * sshd_config(5): add HostKeyAlgorithms option to control which
   public key types are offered for host authentications.

 * ssh(1), sshd(8): extend Ciphers, MACs, KexAlgorithms,
   HostKeyAlgorithms, PubkeyAcceptedKeyTypes and HostbasedKeyTypes
   options to allow appending to the default set of algorithms
   instead of replacing it. Options may now be prefixed with a '+'
   to append to the default, e.g. "HostKeyAlgorithms=+ssh-dss".

 * sshd_config(5): PermitRootLogin now accepts an argument of
   'prohibit-password' as a less-ambiguous synonym of 'without-
   password'.

Bugfixes
--------

 * ssh(1), sshd(8): add compatability workarounds for Cisco and more
   PuTTY versions. bz#2424

 * Fix some omissions and errors in the PROTOCOL and PROTOCOL.mux
   documentation relating to Unix domain socket forwarding;
   bz#2421 bz#2422

 * ssh(1): Improve the ssh(1) manual page to include a better
   description of Unix domain socket forwarding; bz#2423

 * ssh(1), ssh-agent(1): skip uninitialised PKCS#11 slots, fixing
   failures to load keys when they are present. bz#2427

 * ssh(1), ssh-agent(1): do not ignore PKCS#11 hosted keys that wth
   empty CKA_ID; bz#2429

 * sshd(8): clarify documentation for UseDNS option; bz#2045

Portable OpenSSH
----------------

 * Check realpath(3) behaviour matches what sftp-server requires and
   use a replacement if necessary.

Checksums:
==========

 - SHA1 (openssh-7.0.tar.gz) = a19ff0bad2a67348b1d01a38a9580236120b7099
 - SHA256 (openssh-7.0.tar.gz) = 4F6HV/ZqT465f3sMB2vIkXO+wrYtL5hnqzAymfbZ1Jk=

 - SHA1 (openssh-7.0p1.tar.gz) = d8337c9eab91d360d104f6dd805f8b32089c063c
 - SHA256 (openssh-7.0p1.tar.gz) = /VkySToZ9MgRU9gS7k4EK0m707dZqz2TRKvswrwUheU=

Please note that the PGP key used to sign releases was recently rotated.
The new key has been signed by the old key to provide continuity. It is
available from the mirror sites as RELEASE_KEY.asc.

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 6.9/6.9p1 (2015-07-01)

OpenSSH 6.9 was released on 2015-07-01. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
http://www.openssh.com/donations.html

Future Deprecation Notice
=========================

The 7.0 release of OpenSSH, due for release in late July, will
deprecate several features, some of which may affect compatibility
or existing configurations. The intended changes are as follows:

 * The default for the sshd_config(5) PermitRootLogin option will
   change from "yes" to "no".

 * Support for the legacy version 1.x of the SSH protocol will be
   disabled at compile time by default.

 * Support for the 1024-bit diffie-hellman-group1-sha1 key exchange
   will be run-time disabled by default.

 * Support for ssh-dss, ssh-dss-cert-* host and user keys will be
   run-time disabled by default.

 * Support for the legacy v00 cert format will be removed

 * Several ciphers will be disabled by default: blowfish-cbc,
   cast128-cbc, all arcfour variants and the rijndael-cbc aliases
   for AES

 * Refusing all RSA keys smaller than 1024 bits (the current minimum
   is 768 bits)

This list reflects our current intentions, but please check the final
release notes for OpenSSH 7.0 when it is released.

Changes since OpenSSH 6.8
=========================

This is primarily a bugfix release.

Security
----

 * ssh(1): when forwarding X11 connections with ForwardX11Trusted=no,
   connections made after ForwardX11Timeout expired could be permitted
   and no longer subject to XSECURITY restrictions because of an
   ineffective timeout check in ssh(1) coupled with "fail open"
   behaviour in the X11 server when clients attempted connections with
   expired credentials. This problem was reported by Jann Horn.

 * ssh-agent(1): fix weakness of agent locking (ssh-add -x) to
   password guessing by implementing an increasing failure delay,
   storing a salted hash of the password rather than the password
   itself and using a timing-safe comparison function for verifying
   unlock attempts. This problem was reported by Ryan Castellucci.

New Features
----

 * ssh(1), sshd(8): promote chacha20-poly1305@openssh.com to be the
   default cipher

 * sshd(8): support admin-specified arguments to AuthorizedKeysCommand;
   bz#2081

 * sshd(8): add AuthorizedPrincipalsCommand that allows retrieving
   authorized principals information from a subprocess rather than
   a file.

 * ssh(1), ssh-add(1): support PKCS#11 devices with external PIN
   entry devices bz#2240

 * sshd(8): allow GSSAPI host credential check to be relaxed for
   multihomed hosts via GSSAPIStrictAcceptorCheck option; bz#928

 * ssh-keygen(1): support "ssh-keygen -lF hostname" to search
   known_hosts and print key hashes rather than full keys.

 * ssh-agent(1): add -D flag to leave ssh-agent in foreground without
   enabling debug mode; bz#2381

Bugfixes
----

 * ssh(1), sshd(8): deprecate legacy SSH2_MSG_KEX_DH_GEX_REQUEST_OLD
   message and do not try to use it against some 3rd-party SSH
   implementations that use it (older PuTTY, WinSCP).

 * Many fixes for problems caused by compile-time deactivation of
   SSH1 support (including bz#2369)

 * ssh(1), sshd(8): cap DH-GEX group size at 4Kbits for Cisco
   implementations as some would fail when attempting to use group
   sizes >4K; bz#2209

 * ssh(1): fix out-of-bound read in EscapeChar configuration option
   parsing; bz#2396

 * sshd(8): fix application of PermitTunnel, LoginGraceTime,
   AuthenticationMethods and StreamLocalBindMask options in Match
   blocks

 * ssh(1), sshd(8): improve disconnection message on TCP reset;
   bz#2257

 * ssh(1): remove failed remote forwards established by muliplexing
   from the list of active forwards; bz#2363

 * sshd(8): make parsing of authorized_keys "environment=" options
   independent of PermitUserEnv being enabled; bz#2329

 * sshd(8): fix post-auth crash with permitopen=none; bz#2355

 * ssh(1), ssh-add(1), ssh-keygen(1): allow new-format private keys
   to be encrypted with AEAD ciphers; bz#2366

 * ssh(1): allow ListenAddress, Port and AddressFamily configuration
   options to appear in any order; bz#86

 * sshd(8): check for and reject missing arguments for VersionAddendum
   and ForceCommand; bz#2281

 * ssh(1), sshd(8): don't treat unknown certificate extensions as
   fatal; bz#2387

 * ssh-keygen(1): make stdout and stderr output consistent; bz#2325

 * ssh(1): mention missing DISPLAY environment in debug log when X11
   forwarding requested; bz#1682

 * sshd(8): correctly record login when UseLogin is set; bz#378

 * sshd(8): Add some missing options to sshd -T output and fix output
   of VersionAddendum and HostCertificate. bz#2346

 * Document and improve consistency of options that accept a "none"
   argument" TrustedUserCAKeys, RevokedKeys (bz#2382),
   AuthorizedPrincipalsFile (bz#2288)

 * ssh(1): include remote username in debug output; bz#2368

 * sshd(8): avoid compatibility problem with some versions of Tera
   Term, which would crash when they received the hostkeys notification
   message (hostkeys-00@openssh.com)

 * sshd(8): mention ssh-keygen -E as useful when comparing legacy MD5
   host key fingerprints; bz#2332

 * ssh(1): clarify pseudo-terminal request behaviour and use make
   manual language consistent; bz#1716

 * ssh(1): document that the TERM environment variable is not subject
   to SendEnv and AcceptEnv; bz#2386

Portable OpenSSH
----

 * sshd(8): Format UsePAM setting when using sshd -T, part of bz#2346

 * Look for '${host}-ar' before 'ar', making cross-compilation easier;
   bz#2352.

 * Several portable compilation fixes: bz#2402, bz#2337, bz#2370

 * moduli(5): update DH-GEX moduli

Checksums:
==========

 - SHA1 (openssh-6.9.tar.gz) = cd5fcb93411025bbc4b4b57753b622769dfb1e0d
 - SHA256 (openssh-6.9.tar.gz) = itCMw0aE/xvrGKWhzRD2UM/9kzIOyFaH2dIWMfX8agQ=

 - SHA1 (openssh-6.9p1.tar.gz) = 86ab57f00d0fd9bf302760f2f6deac1b6e9df265
 - SHA256 (openssh-6.9p1.tar.gz) = bgdN9TjzV9RAvmz5PcWBoh8i054jbyF/zY6su2yJbP4=

Please note that the PGP key used to sign releases was recently rotated.
The new key has been signed by the old key to provide continuity. It is
available from the mirror sites as RELEASE_KEY.asc.

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.

OpenSSH 6.8/6.8p1 (2015-03-18)

```
OpenSSH 6.8 was released on 2015-03-18. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
http://www.openssh.com/donations.html

Changes since OpenSSH 6.7
=========================

This is a major release, containing a number of new features as
well as a large internal re-factoring.

Potentially-incompatible changes
--------------------------------

 * sshd(8): UseDNS now defaults to 'no'. Configurations that match
   against the client host name (via sshd_config or authorized_keys)
   may need to re-enable it or convert to matching against addresses.

New Features
------------

 * Much of OpenSSH's internal code has been re-factored to be more
   library-like. These changes are mostly not user-visible, but
   have greatly improved OpenSSH's testability and internal layout.

 * Add FingerprintHash option to ssh(1) and sshd(8), and equivalent
   command-line flags to the other tools to control algorithm used
   for key fingerprints. The default changes from MD5 to SHA256 and
   format from hex to base64.

   Fingerprints now have the hash algorithm prepended. An example of
   the new format: SHA256:mVPwvezndPv/ARoIadVY98vAC0g+P/5633yTC4d/wXE
   Please note that visual host keys will also be different.

 * ssh(1), sshd(8): Experimental host key rotation support. Add a
   protocol extension for a server to inform a client of all its
   available host keys after authentication has completed. The client
   may record the keys in known_hosts, allowing it to upgrade to better
   host key algorithms and a server to gracefully rotate its keys.

   The client side of this is controlled by a UpdateHostkeys config
   option (default off).

 * ssh(1): Add a ssh_config HostbasedKeyType option to control which
   host public key types are tried during host-based authentication.

 * ssh(1), sshd(8): fix connection-killing host key mismatch errors
   when sshd offers multiple ECDSA keys of different lengths.

 * ssh(1): when host name canonicalisation is enabled, try to
   parse host names as addresses before looking them up for
   canonicalisation. fixes bz#2074 and avoiding needless DNS
   lookups in some cases.

 * ssh-keygen(1), sshd(8): Key Revocation Lists (KRLs) no longer
   require OpenSSH to be compiled with OpenSSL support.

 * ssh(1), ssh-keysign(8): Make ed25519 keys work for host based
   authentication.

 * sshd(8): SSH protocol v.1 workaround for the Meyer, et al,
   Bleichenbacher Side Channel Attack. Fake up a bignum key before
   RSA decryption.

 * sshd(8): Remember which public keys have been used for
   authentication and refuse to accept previously-used keys.
   This allows AuthenticationMethods=publickey,publickey to require
   that users authenticate using two _different_ public keys.

 * sshd(8): add sshd_config HostbasedAcceptedKeyTypes and
   PubkeyAcceptedKeyTypes options to allow sshd to control what
   public key types will be accepted. Currently defaults to all.

 * sshd(8): Don't count partial authentication success as a failure
   against MaxAuthTries.

 * ssh(1): Add RevokedHostKeys option for the client to allow
   text-file or KRL-based revocation of host keys.

 * ssh-keygen(1), sshd(8): Permit KRLs that revoke certificates by
   serial number or key ID without scoping to a particular CA.

 * ssh(1): Add a "Match canonical" criteria that allows ssh_config
   Match blocks to trigger only in the second config pass.

 * ssh(1): Add a -G option to ssh that causes it to parse its
   configuration and dump the result to stdout, similar to "sshd -T".
```

```
 * ssh(1): Allow Match criteria to be negated. E.g. "Match !host".

 * The regression test suite has been extended to cover more OpenSSH
   features. The unit tests have been expanded and now cover key
   exchange.

Bugfixes

 * ssh-keyscan(1): ssh-keyscan has been made much more robust again
   servers that hang or violate the SSH protocol.

 * ssh(1), ssh-keygen(1): Fix regression bz#2306: Key path names were
   being lost as comment fields.

 * ssh(1): Allow ssh_config Port options set in the second config
   parse phase to be applied (they were being ignored). bz#2286

 * ssh(1): Tweak config re-parsing with host canonicalisation - make
   the second pass through the config files always run when host name
   canonicalisation is enabled (and not whenever the host name
   changes) bz#2267

 * ssh(1): Fix passing of wildcard forward bind addresses when
   connection multiplexing is in use; bz#2324;

 * ssh-keygen(1): Fix broken private key conversion from non-OpenSSH
   formats; bz#2345.

 * ssh-keygen(1): Fix KRL generation bug when multiple CAs are in
   use.

 * Various fixes to manual pages: bz#2288, bz#2316, bz#2273

Portable OpenSSH

 * Support --without-openssl at configure time

   Disables and removes dependency on OpenSSL. Many features,
   including SSH protocol 1 are not supported and the set of crypto
   options is greatly restricted. This will only work on systems
   with native arc4random or /dev/urandom.

   Considered highly experimental for now.

 * Support --without-ssh1 option at configure time

   Allows disabling support for SSH protocol 1.

 * sshd(8): Fix compilation on systems with IPv6 support in utmpx; bz#2296

 * Allow custom service name for sshd on Cygwin. Permits the use of
   multiple sshd running with different service names.

Checksums:
==========

 - SHA1 (openssh-6.8.tar.gz) = 99903c6ca76e0a2c044711017f81127e12459d37
 - SHA256 (openssh-6.8.tar.gz) = N1uzVarFbrm2CzAwuDu3sRoszmqpK+5phAChP/QNyuw=

 - SHA1 (openssh-6.8p1.tar.gz) = cdbc51e46a902b30d263b05fdc71340920e91c92
 - SHA256 (openssh-6.8p1.tar.gz) = P/ZM5z7hJEgLW/dnuYMNfTwDu8tqvnFrePAZLDfOFg4=

Please note that the PGP key used to sign releases was recently rotated.
The new key has been signed by the old key to provide continuity. It is
available from the mirror sites as RELEASE_KEY.asc.

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 6.7/6.7p1 (2014-10-06)

```
OpenSSH 6.7 was released on 2014-10-06. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
http://www.openssh.com/donations.html

Changes since OpenSSH 6.6
=========================

Potentially-incompatible changes

 * sshd(8): The default set of ciphers and MACs has been altered to
```

remove unsafe algorithms. In particular, CBC ciphers and arcfour* are disabled by default.

The full set of algorithms remains available if configured explicitly via the Ciphers and MACs sshd_config options.

* sshd(8): Support for tcpwrappers/libwrap has been removed.

* OpenSSH 6.5 and 6.6 have a bug that causes ~0.2% of connections using the curve25519-sha256@libssh.org KEX exchange method to fail when connecting with something that implements the specification correctly. OpenSSH 6.7 disables this KEX method when speaking to one of the affected versions.

New Features

* Major internal refactoring to begin to make part of OpenSSH usable as a library. So far the wire parsing, key handling and KRL code has been refactored. Please note that we do not consider the API stable yet, nor do we offer the library in separable form.

* ssh(1), sshd(8): Add support for Unix domain socket forwarding. A remote TCP port may be forwarded to a local Unix domain socket and vice versa or both ends may be a Unix domain socket.

* ssh(1), ssh-keygen(1): Add support for SSHFP DNS records for ED25519 key types.

* sftp(1): Allow resumption of interrupted uploads.

* ssh(1): When rekeying, skip file/DNS lookups of the hostkey if it is the same as the one sent during initial key exchange; bz#2154

* sshd(8): Allow explicit ::1 and 127.0.0.1 forwarding bind addresses when GatewayPorts=no; allows client to choose address family; bz#2222

* sshd(8): Add a sshd_config PermitUserRC option to control whether ~/.ssh/rc is executed, mirroring the no-user-rc authorized_keys option; bz#2160

* ssh(1): Add a %C escape sequence for LocalCommand and ControlPath that expands to a unique identifer based on a hash of the tuple of (local host, remote user, hostname, port). Helps avoid exceeding miserly pathname limits for Unix domain sockets in multiplexing control paths; bz#2220

* sshd(8): Make the "Too many authentication failures" message include the user, source address, port and protocol in a format similar to the authentication success / failure messages; bz#2199

* Added unit and fuzz tests for refactored code. These are run automatically in portable OpenSSH via the "make tests" target.

Bugfixes

* sshd(8): Fix remote forwarding with the same listen port but different listen address.

* ssh(1): Fix inverted test that caused PKCS#11 keys that were explicitly listed in ssh_config or on the commandline not to be preferred.

* ssh-keygen(1): Fix bug in KRL generation: multiple consecutive revoked certificate serial number ranges could be serialised to an invalid format. Readers of a broken KRL caused by this bug will fail closed, so no should-have-been-revoked key will be accepted.

* ssh(1): Reflect stdio-forward ("ssh -W host:port ...") failures in exit status. Previously we were always returning 0; bz#2255

* ssh(1), ssh-keygen(1): Make Ed25519 keys' title fit properly in the randomart border; bz#2247

* ssh-agent(1): Only cleanup agent socket in the main agent process and not in any subprocesses it may have started (e.g. forked askpass). Fixes agent sockets being zapped when askpass processes fatal(); bz#2236

* ssh-add(1): Make stdout line-buffered; saves partial output getting lost when ssh-add fatal()s part-way through (e.g. when listing keys from an agent that supports key types that ssh-add doesn't); bz#2234

* ssh-keygen(1): When hashing or removing hosts, don't choke on @revoked markers and don't remove @cert-authority markers; bz#2241

* ssh(1): Don't fatal when hostname canonicalisation fails and a ProxyCommand is in use; continue and allow the ProxyCommand to connect anyway (e.g. to a host with a name outside the DNS behind a bastion)

* scp(1): When copying local->remote fails during read, don't send uninitialised heap to the remote end.

* sftp(1): Fix fatal "el_insertstr failed" errors when tab-completing filenames with a single quote char somewhere in the string; bz#2238

```
 * ssh-keyscan(1): Scan for Ed25519 keys by default.

 * ssh(1): When using VerifyHostKeyDNS with a DNSSEC resolver, down-
   convert any certificate keys to plain keys and attempt SSHFP
   resolution.  Prevents a server from skipping SSHFP lookup and
   forcing a new-hostkey dialog by offering only certificate keys.

 * sshd(8): Avoid crash at exit via NULL pointer reference; bz#2225

 * Fix some strict-alignment errors.

Portable OpenSSH

 * Portable OpenSSH now supports building against libressl-portable.

 * Portable OpenSSH now requires openssl 0.9.8f or greater. Older
   versions are no longer supported.

 * In the OpenSSL version check, allow fix version upgrades (but not
   downgrades. Debian bug #748150.

 * sshd(8): On Cygwin, determine privilege separation user at runtime,
   since it may need to be a domain account.

 * sshd(8): Don't attempt to use vhangup on Linux. It doesn't work for
   non-root users, and for them it just messes up the tty settings.

 * Use CLOCK_BOOTTIME in preference to CLOCK_MONOTONIC when it is
   available. It considers time spent suspended, thereby ensuring
   timeouts (e.g. for expiring agent keys) fire correctly.  bz#2228

 * Add support for ed25519 to opensshd.init init script.

 * sftp-server(8): On platforms that support it, use prctl() to
   prevent sftp-server from accessing /proc/self/{mem,maps}

Checksums:
==========

 - SHA1 (openssh-6.7.tar.gz) = 315497b27a0186e4aef67987cfc9f3d9ba561cd8
 - SHA256 (openssh-6.7.tar.gz) = /me/hPxDw9Tfd3siNKQubSQph84qiKwftiMsgj6nh5E=

 - SHA1 (openssh-6.7p1.tar.gz) = 14e5fbed710ade334d65925e080d1aaeb9c85bf6
 - SHA256 (openssh-6.7p1.tar.gz) = svg5Tq6Fjau9732sELma7ADJVGJ1PoA0LlMLu29yVQc=

Please note that the PGP key used to sign releases was recently rotated.
The new key has been signed by the old key to provide continuity. It is
available from the mirror sites as RELEASE_KEY.asc.

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 6.6/6.6p1 (2014-03-15)

```
OpenSSH 6.6 was released on 2014-03-15. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
http://www.openssh.com/donations.html

Changes since OpenSSH 6.6
=========================

This is primarily a bugfix release.

Security:

 * sshd(8): when using environment passing with a sshd_config(5)
   AcceptEnv pattern with a wildcard. OpenSSH prior to 6.6 could be
   tricked into accepting any enviornment variable that contains the
   characters before the wildcard character.

New / changed features:

 * ssh(1), sshd(8): this release removes the J-PAKE authentication code.
   This code was experimental, never enabled and had been unmaintained
   for some time.

 * ssh(1): when processing Match blocks, skip 'exec' clauses other clauses
   predicates failed to match.

 * ssh(1): if hostname canonicalisation is enabled and results in the
```

```
    destination hostname being changed, then re-parse ssh_config(5) files
    using the new destination hostname. This gives 'Host' and 'Match'
    directives that use the expanded hostname a chance to be applied.

Bugfixes:

 * ssh(1): avoid spurious "getsockname failed: Bad file descriptor" in
   ssh -W. bz#2200, debian#738692

 * sshd(8): allow the shutdown(2) syscall in seccomp-bpf and systrace
   sandbox modes, as it is reachable if the connection is terminated
   during the pre-auth phase.

 * ssh(1), sshd(8): fix unsigned overflow that in SSH protocol 1 bignum
   parsing. Minimum key length checks render this bug unexploitable to
   compromise SSH 1 sessions.

 * sshd_config(5): clarify behaviour of a keyword that appears in
   multiple matching Match blocks. bz#2184

 * ssh(1): avoid unnecessary hostname lookups when canonicalisation is
   disabled. bz#2205

 * sshd(8): avoid sandbox violation crashes in GSSAPI code by caching
   the supported list of GSSAPI mechanism OIDs before entering the
   sandbox. bz#2107

 * ssh(1): fix possible crashes in SOCKS4 parsing caused by assumption
   that the SOCKS username is nul-terminated.

 * ssh(1): fix regression for UsePrivilegedPort=yes when BindAddress is
   not specified.

 * ssh(1), sshd(8): fix memory leak in ECDSA signature verification.

 * ssh(1): fix matching of 'Host' directives in ssh_config(5) files
   to be case-insensitive again (regression in 6.5).

Portable OpenSSH:

 * sshd(8): don't fatal if the FreeBSD Capsicum is offered by the
   system headers and libc but is not supported by the kernel.
 * Fix build using the HP-UX compiler.

Checksums:
==========

 - SHA1 (openssh-6.6.tar.gz) = bf932d798324ff2502409d3714d0ad8d65c7e1e7
 - SHA256 (openssh-6.6.tar.gz) = jaSJE5aiQRm+91dV6EvVGr/ozo33tbxyjjFSiu+Cy80=

 - SHA1 (openssh-6.6p1.tar.gz) = b850fd1af704942d9b3c2eff7ef6b3a59b6a6b6e
 - SHA256 (openssh-6.6p1.tar.gz) = SMHwZktFNIdQOABMxPNVW4MpwqgcHfSNtcUXgA3iA7s=

Please note that the PGP key used to sign releases was recently rotated.
The new key has been signed by the old key to provide continuity. It is
available from the mirror sites as RELEASE_KEY.asc.

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 6.5/6.5p1 (2014-01-30)

```
Changes since OpenSSH 6.4
=========================

This is a feature-focused release.

New features:

 * ssh(1), sshd(8): Add support for key exchange using elliptic-curve
   Diffie Hellman in Daniel Bernstein's Curve25519. This key exchange
   method is the default when both the client and server support it.

 * ssh(1), sshd(8): Add support for Ed25519 as a public key type.
   Ed25519 is a elliptic curve signature scheme that offers
   better security than ECDSA and DSA and good performance. It may be
   used for both user and host keys.

 * Add a new private key format that uses a bcrypt KDF to better
   protect keys at rest. This format is used unconditionally for
   Ed25519 keys, but may be requested when generating or saving
   existing keys of other types via the -o ssh-keygen(1) option.
   We intend to make the new format the default in the near future.
   Details of the new format are in the PROTOCOL.key file.

 * ssh(1), sshd(8): Add a new transport cipher
   "chacha20-poly1305@openssh.com" that combines Daniel Bernstein's
   ChaCha20 stream cipher and Poly1305 MAC to build an authenticated
```

encryption mode. Details are in the PROTOCOL.chacha20poly1305 file.

* ssh(1), sshd(8): Refuse RSA keys from old proprietary clients and servers that use the obsolete RSA+MD5 signature scheme. It will still be possible to connect with these clients/servers but only DSA keys will be accepted, and OpenSSH will refuse connection entirely in a future release.

* ssh(1), sshd(8): Refuse old proprietary clients and servers that use a weaker key exchange hash calculation.

* ssh(1): Increase the size of the Diffie-Hellman groups requested for each symmetric key size. New values from NIST Special Publication 800-57 with the upper limit specified by RFC4419.

* ssh(1), ssh-agent(1): Support PKCS#11 tokens that only provide X.509 certs instead of raw public keys (requested as bz#1908).

* ssh(1): Add a ssh_config(5) "Match" keyword that allows conditional configuration to be applied by matching on hostname, user and result of arbitrary commands.

* ssh(1): Add support for client-side hostname canonicalisation using a set of DNS suffixes and rules in ssh_config(5). This allows unqualified names to be canonicalised to fully-qualified domain names to eliminate ambiguity when looking up keys in known_hosts or checking host certificate names.

* sftp-server(8): Add the ability to whitelist and/or blacklist sftp protocol requests by name.

* sftp-server(8): Add a sftp "fsync@openssh.com" to support calling fsync(2) on an open file handle.

* sshd(8): Add a ssh_config(5) PermitTTY to disallow TTY allocation, mirroring the longstanding no-pty authorized_keys option.

* ssh(1): Add a ssh_config ProxyUseFDPass option that supports the use of ProxyCommands that establish a connection and then pass a connected file descriptor back to ssh(1). This allows the ProxyCommand to exit rather than staying around to transfer data.

Bugfixes:

* ssh(1), sshd(8): Fix potential stack exhaustion caused by nested certificates.

* ssh(1): bz#1211: make BindAddress work with UsePrivilegedPort.

* sftp(1): bz#2137: fix the progress meter for resumed transfer.

* ssh-add(1): bz#2187: do not request smartcard PIN when removing keys from ssh-agent.

* sshd(8): bz#2139: fix re-exec fallback when original sshd binary cannot be executed.

* ssh-keygen(1): Make relative-specified certificate expiry times relative to current time and not the validity start time.

* sshd(8): bz#2161: fix AuthorizedKeysCommand inside a Match block.

* sftp(1): bz#2129: symlinking a file would incorrectly canonicalise the target path.

* ssh-agent(1): bz#2175: fix a use-after-free in the PKCS#11 agent helper executable.

* sshd(8): Improve logging of sessions to include the user name, remote host and port, the session type (shell, command, etc.) and allocated TTY (if any).

* sshd(8): bz#1297: tell the client (via a debug message) when their preferred listen address has been overridden by the server's GatewayPorts setting.

* sshd(8): bz#2162: include report port in bad protocol banner message.

* sftp(1): bz#2163: fix memory leak in error path in do_readdir().

* sftp(1): bz#2171: don't leak file descriptor on error.

* sshd(8): Include the local address and port in "Connection from ..." message (only shown at loglevel>=verbose).

Portable OpenSSH:

* Please note that this is the last version of Portable OpenSSH that will support versions of OpenSSL prior to 0.9.6. Support (i.e. SSH_OLD_EVP) will be removed following the 6.5p1 release.

* Portable OpenSSH will attempt compile and link as a Position Independent Executable on Linux, OS X and OpenBSD on recent gcc-like compilers. Other platforms and older/other compilers may request this using the --with-pie configure flag.

* A number of other toolchain-related hardening options are used

```
    automatically if available, including -ftrapv to abort on signed
    integer overflow and options to write-protect dynamic linking
    information.  The use of these options may be disabled using the
    --without-hardening configure flag.

  * If the toolchain supports it, one of the -fstack-protector-strong,
    -fstack-protector-all or -fstack-protector compilation flag are
    used to add guards to mitigate attacks based on stack overflows.
    The use of these options may be disabled using the
    --without-stackprotect configure option.

  * sshd(8): Add support for pre-authentication sandboxing using the
    Capsicum API introduced in FreeBSD 10.

  * Switch to a ChaCha20-based arc4random() PRNG for platforms that do
    not provide their own.

  * sshd(8): bz#2156: restore Linux oom_adj setting when handling
    SIGHUP to maintain behaviour over retart.

  * sshd(8): bz#2032: use local username in krb5_kuserok check rather
    than full client name which may be of form user@REALM.

  * ssh(1), sshd(8): Test for both the presence of ECC NID numbers in
    OpenSSL and that they actually work. Fedora (at least) has
    NID_secp521r1 that doesn't work.

  * bz#2173: use pkg-config --libs to include correct -L location for
    libedit.

Checksums:
==========

 - SHA1 (openssh-6.5.tar.gz) = 0a375e20d895670489a9241f8faa57670214fbed
 - SHA256 (openssh-6.5.tar.gz) = sK5q2rB0o5JCbEmbeE/6N9DtJkT81dwmeuhogT4i900=

 - SHA1 (openssh-6.5p1.tar.gz) = 3363a72b4fee91b29cf2024ff633c17f6cd2f86d
 - SHA256 (openssh-6.5p1.tar.gz) = oRle1V25RSUtWhcw1KKipcHJpqoB7y5a91CpYmI9kCc=

Please note that the PGP key used to sign releases has been rotated.
The new key has been signed by the old key to provide continuity. It
is available from the mirror sites as RELEASE_KEY.asc.

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 6.4/6.4p1 (2013-11-08)

```
Changes since OpenSSH 6.3
=========================

This release fixes a security bug:

 * sshd(8): fix a memory corruption problem triggered during rekeying
   when an AES-GCM cipher is selected. Full details of the vulnerability
   are available at: http://www.openssh.com/txt/gcmrekey.adv

Checksums:
==========

 - SHA1 (openssh-6.4.tar.gz) = 4caf1a50eb3a3da821c16298c4aaa576fe24210c
 - SHA1 (openssh-6.4p1.tar.gz) = cf5fe0eb118d7e4f9296fbc5d6884965885fc55d

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 6.3/6.3p1 (2013-09-13)

```
Changes since OpenSSH 6.2
=========================

This release is predominantly a bugfix release:

Features:

 * sshd(8): add ssh-agent(1) support to sshd(8); allows encrypted hostkeys,
   or hostkeys on smartcards.
```

* ssh(1)/sshd(8): allow optional time-based rekeying via a second argument to the existing RekeyLimit option. RekeyLimit is now supported in sshd_config as well as on the client.

* sshd(8): standardise logging of information during user authentication.

  The presented key/cert and the remote username (if available) is now logged in the authentication success/failure message on the same log line as the local username, remote host/port and protocol in use. Certificates contents and the key fingerprint of the signing CA are logged too.

  Including all relevant information on a single line simplifies log analysis as it is no longer necessary to relate information scattered across multiple log entries.

* ssh(1): add the ability to query which ciphers, MAC algorithms, key types and key exchange methods are supported in the binary.

* ssh(1): support ProxyCommand=- to allow support cases where stdin and stdout already point to the proxy.

* ssh(1): allow IdentityFile=none

* ssh(1)/sshd(8): add -E option to ssh and sshd to append debugging logs to a specified file instead of stderr or syslog.

* sftp(1): add support for resuming partial downloads using the "reget" command and on the sftp commandline or on the "get" commandline using the "-a" (append) option.

* ssh(1): add an "IgnoreUnknown" configuration option to selectively suppress errors arising from unknown configuration directives.

* sshd(8): add support for submethods to be appended to required authentication methods listed via AuthenticationMethods.

Bugfixes:

* sshd(8): fix refusal to accept certificate if a key of a different type to the CA key appeared in authorized_keys before the CA key.

* ssh(1)/ssh-agent(1)/sshd(8): Use a monotonic time source for timers so that things like keepalives and rekeying will work properly over clock steps.

* sftp(1): update progressmeter when data is acknowledged, not when it's sent. bz#2108

* ssh(1)/ssh-keygen(1): improve error messages when the current user does not exist in /etc/passwd; bz#2125

* ssh(1): reset the order in which public keys are tried after partial authentication success.

* ssh-agent(1): clean up socket files after SIGINT when in debug mode; bz#2120

* ssh(1) and others: avoid confusing error messages in the case of broken system resolver configurations; bz#2122

* ssh(1): set TCP nodelay for connections started with -N; bz#2124

* ssh(1): correct manual for permission requirements on ~/.ssh/config; bz#2078

* ssh(1): fix ControlPersist timeout not triggering in cases where TCP connections have hung. bz#1917

* ssh(1): properly deatch a ControlPersist master from its controlling terminal.

* sftp(1): avoid crashes in libedit when it has been compiled with multi-byte character support. bz#1990

* sshd(8): when running sshd -D, close stderr unless we have explicitly requested logging to stderr. bz#1976,

* ssh(1): fix incomplete bzero; bz#2100

* sshd(8): log and error and exit if ChrootDirectory is specified and running without root privileges.

* Many improvements to the regression test suite. In particular log files are now saved from ssh and sshd after failures.

* Fix a number of memory leaks. bz#1967 bz#2096 and others

* sshd(8): fix public key authentication when a :style is appended to the requested username.

* ssh(1): do not fatally exit when attempting to cleanup multiplexing-created channels that are incompletely opened. bz#2079

Portable OpenSSH:

* Major overhaul of contrib/cygwin/README

```
 * Fix unaligned accesses in umac.c for strict-alignment architectures.
   bz#2101

 * Enable -Wsizeof-pointer-memaccess if the compiler supports it. bz#2100

 * Fix broken incorrect commandline reporting errors. bz#1448

 * Only include SHA256 and ECC-based key exchange methods if libcrypto has
   the required support.

 * Fix crash in SOCKS5 dynamic forwarding code on strict-alignment
   architectures.

 * A number of portability fixes for Android:
   * Don't try to use lastlog on Android; bz#2111
   * Fall back to using openssl's DES_crypt function on platorms that don't
     have a native crypt() function; bz#2112
   * Test for fd_mask, howmany and NFDBITS rather than trying to enumerate
     the plaforms that don't have them. bz#2085
   * Replace S_IWRITE, which isn't standardized, with S_IWUSR, which is.
     bz#2085
   * Add a null implementation of endgrent for platforms that don't have
     it (eg Android) bz#2087
   * Support platforms, such as Android, that lack struct passwd.pw_gecos.
     bz#2086

Checksums:
==========

 - SHA1 (openssh-6.3.tar.gz) = 8a6ef99ffc80c19e9afe9fe1e857370f6adcf450
 - SHA1 (openssh-6.3p1.tar.gz) = 70845ca79474258cab29dbefae13d93e41a83ccb

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 6.2p2 (2013-05-16)

```
Changes since OpenSSH 6.2p1
===========================

This is a bugfix release:

Bugfixes:

 * ssh(1): Only warn for missing identity files that were explicitly
   specified.

 * Fix bug in contributed contrib/ssh-copy-id script that could result in
   "rm *" being called on mktemp failure. bz#2105

 * sshd(8): Quiet disconnect notifications on the server from error() back
   to logit() from error() for normal, client-initiated disconnections.
   bz#2057

 * Avoid conflicting definitions of __int64 on Cygwin

Checksums:
==========

 - SHA1 (openssh-6.2p2.tar.gz) = c2b4909eba6f5ec6f9f75866c202db47f3b501ba

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 6.2/6.2p1 (2013-03-22)

```
Changes since OpenSSH 6.1
=========================

This release introduces a number of new features:

Features:

 * ssh(1)/sshd(8): Added support for AES-GCM authenticated encryption in
   SSH protocol 2. The new cipher is available as aes128-gcm@openssh.com
   and aes256-gcm@openssh.com. It uses an identical packet format to the
```

AES-GCM mode specified in RFC 5647, but uses simpler and different selection rules during key exchange.

* ssh(1)/sshd(8): Added support for encrypt-then-mac (EtM) MAC modes for SSH protocol 2. These modes alter the packet format and compute the MAC over the packet length and encrypted packet rather than over the plaintext data. These modes are considered more secure and are used by default when available.

* ssh(1)/sshd(8): Added support for the UMAC-128 MAC as "umac-128@openssh.com" and "umac-128-etm@openssh.com". The latter being an encrypt-then-mac mode.

* sshd(8): Added support for multiple required authentication in SSH protocol 2 via an AuthenticationMethods option. This option lists one or more comma-separated lists of authentication method names. Successful completion of all the methods in any list is required for authentication to complete. This allows, for example, requiring a user having to authenticate via public key or GSSAPI before they are offered password authentication.

* sshd(8)/ssh-keygen(1): Added support for Key Revocation Lists (KRLs), a compact binary format to represent lists of revoked keys and certificates that take as little as one bit per certificate when revoking by serial number. KRLs may be generated using ssh-keygen(1) and are loaded into sshd(8) via the existing RevokedKeys sshd_config option.

* ssh(1): IdentitiesOnly now applies to keys obtained from a PKCS11Provider. This allows control of which keys are offered from tokens using IdentityFile.

* sshd(8): sshd_config(5)'s AllowTcpForwarding now accepts "local" and "remote" in addition to its previous "yes"/"no" keywords to allow the server to specify whether just local or remote TCP forwarding is enabled.

* sshd(8): Added a sshd_config(5) option AuthorizedKeysCommand to support fetching authorized_keys from a command in addition to (or instead of) from the filesystem. The command is run under an account specified by an AuthorizedKeysCommandUser sshd_config(5) option.

* sftp-server(8): Now supports a -d option to allow the starting directory to be something other than the user's home directory.

* ssh-keygen(1): Now allows fingerprinting of keys hosted in PKCS#11 tokens using "ssh-keygen -lD pkcs11_provider".

* ssh(1): When SSH protocol 2 only is selected (the default), ssh(1) now immediately sends its SSH protocol banner to the server without waiting to receive the server's banner, saving time when connecting.

* ssh(1): Added ~v and ~V escape sequences to raise and lower the logging level respectively.

* ssh(1): Made the escape command help (~?) context sensitive so that only commands that will work in the current session are shown.

* ssh-keygen(1): When deleting host lines from known_hosts using "ssh-keygen -R host", ssh-keygen(1) now prints details of which lines were removed.

Bugfixes:

* ssh(1): Force a clean shutdown of ControlMaster client sessions when the ~. escape sequence is used. This means that ~. should now work in mux clients even if the server is no longer responding.

* ssh(1): Correctly detect errors during local TCP forward setup in multiplexed clients. bz#2055

* ssh-add(1): Made deleting explicit keys "ssh-add -d" symmetric with adding keys with respect to certificates. It now tries to delete the corresponding certificate and respects the -k option to allow deleting of the key only.

* sftp(1): Fix a number of parsing and command-editing bugs, including bz#1956

* ssh(1): When muxmaster is run with -N, ensured that it shuts down gracefully when a client sends it "-O stop" rather than hanging around. bz#1985

* ssh-keygen(1): When screening moduli candidates, append to the file rather than overwriting to allow resumption. bz#1957

* ssh(1): Record "Received disconnect" messages at ERROR rather than INFO priority. bz#2057.

* ssh(1): Loudly warn if explicitly-provided private key is unreadable. bz#1981

Portable OpenSSH:

* sshd(8): The Linux seccomp-filter sandbox is now supported on ARM platforms where the kernel supports it.

* sshd(8): The seccomp-filter sandbox will not be enabled if the system

```
    headers support it at compile time, regardless of whether it can be
    enabled then. If the run-time system does not support seccomp-filter,
    sshd will fall back to the rlimit pseudo-sandbox.

 * ssh(1): Don't link in the Kerberos libraries. They aren't necessary
   on the client, just on sshd(8). bz#2072

 * Fix GSSAPI linking on Solaris, which uses a differently-named GSSAPI
   library. bz#2073

 * Fix compilation on systems with openssl-1.0.0-fips.

 * Fix a number of errors in the RPM spec files.

Checksums:
==========

 - SHA1 (openssh-6.2.tar.gz) = b3f6cd774d345f22f6d0038cc9464cce131a0676
 - SHA1 (openssh-6.2p1.tar.gz) = 8824708c617cc781b2bb29fa20bd905fd3d2a43d

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 6.1/6.1p1 (2012-08-29)

```
OpenSSH 6.1 was released on 2012-08-29. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
http://www.openssh.com/donations.html

Changes since OpenSSH 6.0
=========================

This is primarily a bugfix release.

Features:

 * sshd(8): This release turns on pre-auth sandboxing sshd by default for
   new installs, by setting UsePrivilegeSeparation=sandbox in sshd_config.
 * ssh-keygen(1): Add options to specify starting line number and number of
   lines to process when screening moduli candidates, allowing processing
   of different parts of a candidate moduli file in parallel
 * sshd(8): The Match directive now supports matching on the local (listen)
   address and port upon which the incoming connection was received via
   LocalAddress and LocalPort clauses.
 * sshd(8): Extend sshd_config Match directive to allow setting AcceptEnv
   and {Allow,Deny}{Users,Groups}
 * Add support for RFC6594 SSHFP DNS records for ECDSA key types. bz#1978
 * ssh-keygen(1): Allow conversion of RSA1 keys to public PEM and PKCS8
 * sshd(8): Allow the sshd_config PermitOpen directive to accept "none" as
   an argument to refuse all port-forwarding requests.
 * sshd(8): Support "none" as an argument for AuthorizedPrincipalsFile
 * ssh-keyscan(1): Look for ECDSA keys by default. bz#1971
 * sshd(8): Add "VersionAddendum" to sshd_config to allow server operators
   to append some arbitrary text to the server SSH protocol banner.

Bugfixes:

 * ssh(1)/sshd(8): Don't spin in accept() in situations of file
   descriptor exhaustion. Instead back off for a while.
 * ssh(1)/sshd(8): Remove hmac-sha2-256-96 and hmac-sha2-512-96 MACs as
   they were removed from the specification. bz#2023,
 * sshd(8): Handle long comments in config files better. bz#2025
 * ssh(1): Delay setting tty_flag so RequestTTY options are correctly
   picked up. bz#1995
 * sshd(8): Fix handling of /etc/nologin incorrectly being applied to root
   on platforms that use login_cap.

Portable OpenSSH:

 * sshd(8): Allow sshd pre-auth sandboxing to fall-back to the rlimit
   sandbox from the Linux SECCOMP filter sandbox when the latter is
   not available in the kernel.
 * ssh(1): Fix NULL dereference when built with LDNS and using DNSSEC to
   retrieve a CNAME SSHFP record.
 * Fix cross-compilation problems related to pkg-config. bz#1996

Checksums:
==========

 - SHA1 (openssh-6.1.tar.gz) = 7ed5b491cfebcaee2273d1f872314107273c2167
 - SHA1 (openssh-6.1p1.tar.gz) = 751c92c912310c3aa9cadc113e14458f843fc7b3
```

```
Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 6.0/6.0p1 (2012-04-22)

```
OpenSSH 6.0 was released on 2012-04-22. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
http://www.openssh.com/donations.html

Changes since OpenSSH 5.9
=========================

This is primarily a bugfix release.

Features:

 * ssh-keygen(1): Add optional checkpoints for moduli screening
 * ssh-add(1): new -k option to load plain keys (skipping certificates)
 * sshd(8): Add wildcard support to PermitOpen, allowing things like
   "PermitOpen localhost:*".  bz #1857
 * ssh(1): support for cancelling local and remote port forwards via the
   multiplex socket. Use ssh -O cancel -L xx:xx:xx -R yy:yy:yy user@host"
   to request the cancellation of the specified forwardings
 * support cancellation of local/dynamic forwardings from ~C commandline

Bugfixes:

 * ssh(1): ensure that $DISPLAY contains only valid characters before
   using it to extract xauth data so that it can't be used to play local
   shell metacharacter games.
 * ssh(1): unbreak remote portforwarding with dynamic allocated listen ports
 * scp(1): uppress adding '--' to remote commandlines when the first
   argument does not start with '-'. saves breakage on some
   difficult-to-upgrade embedded/router platforms
 * ssh(1)/sshd(8): fix typo in IPQoS parsing: there is no "AF14" class,
   but there is an "AF21" class
 * ssh(1)/sshd(8): do not permit SSH2_MSG_SERVICE_REQUEST/ACCEPT during
   rekeying
 * ssh(1): skip attempting to create ~/.ssh when -F is passed
 * sshd(8): unbreak stdio forwarding when ControlPersist is in use; bz#1943
 * sshd(1): send tty break to pty master instead of (probably already
   closed) slave side; bz#1859
 * sftp(1): silence error spam for "ls */foo" in directory with files;
   bz#1683
 * Fixed a number of memory and file descriptor leaks

Portable OpenSSH:

 * Add a new privilege separation sandbox implementation for Linux's
   new seccomp sandbox, automatically enabled on platforms that support
   it. (Note: privilege separation sandboxing is still experimental)
 * Fix compilation problems on FreeBSD, where libutil contained openpty()
   but not login().
 * ssh-keygen(1): don't fail in -A on platforms that don't support ECC
 * Add optional support for LDNS, a BSD licensed DNS resolver library
   which supports DNSSEC
 * Relax OpenSSL version check to allow running OpenSSH binaries on
   systems with OpenSSL libraries with a newer "fix" or "patch" level
   than the binaries were originally compiled on (previous check only
   allowed movement within "patch" releases). bz#1991
 * Fix builds using contributed Redhat spec file. bz#1992

Checksums:
==========

 - SHA1 (openssh-6.0.tar.gz) = 5d30aba0423c44e89924bb44c5d2153635506a9f
 - SHA1 (openssh-6.0p1.tar.gz) = f691e53ef83417031a2854b8b1b661c9c08e4422

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 5.9/5.9p1 (2011-09-06)

```
OpenSSH 5.9 was released on 2011-09-06. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
http://www.openssh.com/donations.html

Changes since OpenSSH 5.8
=========================

Features:

 * Introduce sandboxing of the pre-auth privsep child using an optional
   sshd_config(5) "UsePrivilegeSeparation=sandbox" mode that enables
   mandatory restrictions on the syscalls the privsep child can perform.
   This intention is to prevent a compromised privsep child from being
   used to attack other hosts (by opening sockets and proxying) or
   probing local kernel attack surface.

   Three concrete sandbox implementation are provided (selected at
   configure time): systrace, seatbelt and rlimit.

   The systrace sandbox uses systrace(4) in unsupervised "fast-path"
   mode, where a list of permitted syscalls is supplied. Any syscall not
   on the list results in SIGKILL being sent to the privsep child. Note
   that this requires a kernel with the new SYSTR_POLICY_KILL option
   (only OpenBSD has this mode at present).

   The seatbelt sandbox uses OS X/Darwin sandbox(7) facilities with a
   strict (kSBXProfilePureComputation) policy that disables access to
   filesystem and network resources.

   The rlimit sandbox is a fallback choice for platforms that don't
   support a better one; it uses setrlimit() to reset the hard-limit
   of file descriptors and processes to zero, which should prevent
   the privsep child from forking or opening new network connections.

   Sandboxing of the privilege separated child process is currently
   experimental but should become the default in a future release.
   Native sandboxes for other platforms are welcome (e.g. Capsicum,
   Linux pid/net namespaces, etc.)

 * Add new SHA256-based HMAC transport integrity modes from
   http://www.ietf.org/id/draft-dbider-sha2-mac-for-ssh-02.txt
   These modes are hmac-sha2-256, hmac-sha2-256-96, hmac-sha2-512,
   and hmac-sha2-512-96, and are available by default in ssh(1) and
   sshd(8)

 * The pre-authentication sshd(8) privilege separation slave process
   now logs via a socket shared with the master process, avoiding the
   need to maintain /dev/log inside the chroot.

 * ssh(1) now warns when a server refuses X11 forwarding

 * sshd_config(5)'s AuthorizedKeysFile now accepts multiple paths,
   separated by whitespace. The undocumented AuthorizedKeysFile2
   option is deprecated (though the default for AuthorizedKeysFile
   includes .ssh/authorized_keys2)

 * sshd_config(5): similarly deprecate UserKnownHostsFile2 and
   GlobalKnownHostsFile2 by making UserKnownHostsFile and
   GlobalKnownHostsFile accept multiple options and default to
   include known_hosts2

 * Retain key comments when loading v.2 keys. These will be visible
   in "ssh-add -l" and other places. bz#439

 * ssh(1) and sshd(8): set IPv6 traffic class from IPQoS (as well as
   IPv4 ToS/DSCP). bz#1855

 * ssh_config(5)'s ControlPath option now expands %L to the host
   portion of the destination host name.

 * ssh_config(5) "Host" options now support negated Host matching, e.g.

     Host *.example.org !c.example.org
        User mekmitasdigoat

   Will match "a.example.org", "b.example.org", but not "c.example.org"

 * ssh_config(5): a new RequestTTY option provides control over when a
   TTY is requested for a connection, similar to the existing -t/-tt/-T
   ssh(1) commandline options.

 * sshd(8): allow GSSAPI authentication to detect when a server-side
   failure causes authentication failure and don't count such failures
   against MaxAuthTries; bz#1244

 * ssh-keygen(1): Add -A option. For each of the key types (rsa1, rsa,
   dsa and ecdsa) for which host keys do not exist, generate the host
   keys with the default key file path, an empty passphrase, default
```

```
    bits for the key type, and default comment. This is useful for
    system initialisation scripts.

  * ssh(1): Allow graceful shutdown of multiplexing: request that a mux
    server removes its listener socket and refuse future multiplexing
    requests but don't kill existing connections. This may be requested
    using "ssh -O stop ..."

  * ssh-add(1) now accepts keys piped from standard input. E.g.
    "ssh-add - < /path/to/key"

  * ssh-keysign(8) now signs hostbased authentication
    challenges correctly using ECDSA keys; bz#1858

  * sftp(1): document that sftp accepts square brackets to delimit
    addresses (useful for IPv6); bz#1847a

  * ssh(1): when using session multiplexing, the master process will
    change its process title to reflect the control path in use and
    when a ControlPersist-ed master is waiting to close; bz#1883 and
    bz#1911

  * Other minor bugs fixed: 1849 1861 1862 1869 1875 1878 1879 1892
    1900 1905 1913

Portable OpenSSH Bugfixes:

  * Fix a compilation error in the SELinux support code. bz#1851

  * This release removes support for ssh-rand-helper. OpenSSH now
    obtains its random numbers directly from OpenSSL or from
    a PRNGd/EGD instance specified at configure time.

  * sshd(8) now resets the SELinux process execution context before
    executing passwd for password changes; bz#1891

  * Since gcc >= 4.x ignores all -Wno-options options, test only the
    corresponding -W-option when trying to determine whether it is
    accepted; bz#1901

  * Add ECDSA key generation to the Cygwin ssh-{host,user}-config
    scripts.

  * Updated .spec and init files for Linux; bz#1920

  * Improved SELinux error messages in context change failures and
    suppress error messages when attempting to change from the
    "unconfined_t" type; bz#1924 bz#1919

  * Fix build errors on platforms without dlopen(); bz#1929

Checksums:
==========

 - SHA1 (openssh-5.9.tar.gz) = bc0cb728bbc394769f9a2ce5b8cd99dc41e12632
 - SHA1 (openssh-5.9p1.tar.gz) = ac4e0055421e9543f0af5da607a72cf5922dcc56

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 5.8p2 (2011-05-03)

```
Portable OpenSSH 5.8p2 was released on 2011-05-03. It will be available
from the mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
http://www.openssh.com/donations.html

Changes since OpenSSH 5.8p1
===========================

Security:

  * Fix local private host key compromise on platforms without host-
    level randomness support (e.g. /dev/random) reported by Tomas Mraz

    On hosts that did not have a randomness source configured in
    OpenSSL and were not configured to use EGD/PRNGd (using the
    --with-prngd-socket configure option), the ssh-rand-helper command
    was being implicitly executed by ssh-keysign with open file
    descriptors to the host private keys. An attacker could use
    ptrace(2) to attach to ssh-rand-helper and exfiltrate the keys.
```

```
    Most modern operating systems are not vulnerable. In particular,
    *BSD, Linux, OS X and Cygwin do not use ssh-rand-helper.

    A full advisory for this issue is available at:
    http://www.openssh.com/txt/portable-keysign-rand-helper.adv

Portable OpenSSH Bugfixes:

 * Fix compilation failure when enabling SELinux support.

 * Revised Cygwin ssh-{host,user}-config that include ECDSA key
   support.

 * Revised Cygwin ssh-host-config to be more thorough in error checking
   and reporting.

Checksums:
==========

 - SHA1 (openssh-5.8p2.tar.gz) = 64798328d310e4f06c9f01228107520adbc8b3e5

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 5.8/5.8p1 (2011-02-04)

```
OpenSSH 5.8 was released on 2011-02-04. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
http://www.openssh.com/donations.html

Changes since OpenSSH 5.7
=========================

Security:

 * Fix vulnerability in legacy certificate signing introduced in
   OpenSSH-5.6 and found by Mateusz Kocielski.

   Legacy certificates signed by OpenSSH 5.6 or 5.7 included data from
   the stack in place of a random nonce field. The contents of the stack
   do not appear to contain private data at this point, but this cannot
   be stated with certainty for all platform, library and compiler
   combinations. In particular, there exists a risk that some bytes from
   the privileged CA key may be accidentally included.

   A full advisory for this issue is available at:
   http://www.openssh.com/txt/legacy-cert.adv

Portable OpenSSH Bugfixes:

 * Fix compilation failure when enableing SELinux support.

 * Do not attempt to call SELinux functions when SELinux is disabled.
   bz#1851

Checksums:
==========

 - SHA1 (openssh-5.8.tar.gz) = 205dece2c8b41c69b082eb65320d359987aae25b
 - SHA1 (openssh-5.8p1.tar.gz) = adebb2faa9aba2a3a3c8b401b2b19677ab53f0de

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 5.7/5.7p1 (2011-01-24)

```
OpenSSH 5.7 was released on 2011-01-24. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.
```

Once again, we would like to thank the OpenSSH community for their continued support of the project, especially those who contributed code or patches, reported bugs, tested snapshots or donated to the project. More information on donations may be found at: http://www.openssh.com/donations.html

Changes since OpenSSH 5.6
=========================

Features:

* Implement Elliptic Curve Cryptography modes for key exchange (ECDH) and host/user keys (ECDSA) as specified by RFC5656. ECDH and ECDSA offer better performance than plain DH and DSA at the same equivalent symmetric key length, as well as much shorter keys.

  Only the mandatory sections of RFC5656 are implemented, specifically the three REQUIRED curves nistp256, nistp384 and nistp521 and only ECDH and ECDSA. Point compression (optional in RFC5656) is NOT implemented.

  Certificate host and user keys using the new ECDSA key types are supported - an ECDSA key may be certified, and an ECDSA key may act as a CA to sign certificates.

  ECDH in a 256 bit curve field is the preferred key agreement algorithm when both the client and server support it. ECDSA host keys are preferred when learning a host's keys for the first time, or can be learned using ssh-keyscan(1).

* sftp(1)/sftp-server(8): add a protocol extension to support a hard link operation. It is available through the "ln" command in the client. The old "ln" behaviour of creating a symlink is available using its "-s" option or through the preexisting "symlink" command

* scp(1): Add a new -3 option to scp: Copies between two remote hosts are transferred through the local host. Without this option the data is copied directly between the two remote hosts.

* ssh(1): automatically order the hostkeys requested by the client based on which hostkeys are already recorded in known_hosts. This avoids hostkey warnings when connecting to servers with new ECDSA keys, since these are now preferred when learning hostkeys for the first time.

* ssh(1)/sshd(8): add a new IPQoS option to specify arbitrary TOS/DSCP/QoS values instead of hardcoding lowdelay/throughput. bz#1733

* sftp(1): the sftp client is now significantly faster at performing directory listings, using OpenBSD glob(3) extensions to preserve the results of stat(3) operations performed in the course of its execution rather than performing expensive round trips to fetch them again afterwards.

* ssh(1): "atomically" create the listening mux socket by binding it on a temporary name and then linking it into position after listen() has succeeded. This allows the mux clients to determine that the server socket is either ready or stale without races. stale server sockets are now automatically removed. (also fixes bz#1711)

* ssh(1)/sshd(8): add a KexAlgorithms knob to the client and server configuration to allow selection of which key exchange methods are used by ssh(1) and sshd(8) and their order of preference.

* sftp(1)/scp(1): factor out bandwidth limiting code from scp(1) into a generic bandwidth limiter that can be attached using the atomicio callback mechanism and use it to add a bandwidth limit option to sftp(1). bz#1147

BugFixes:

* ssh(1)/ssh-agent(1): honour $TMPDIR for client xauth and ssh-agent temporary directories. bz#1809

* ssh(1): avoid NULL deref on receiving a channel request on an unknown or invalid channel; bz#1842

* sshd(8): remove a debug() that pollutes stderr on client connecting to a server in debug mode; bz#1719

* scp(1): pass through ssh command-line flags and options when doing remote-remote transfers, e.g. to enable agent forwarding which is particularly useful in this case; bz#1837

* sftp-server(8): umask should be parsed as octal

* sftp(1): escape '[' in filename tab-completion

* ssh(1): Typo in confirmation message. bz#1827

* sshd(8): prevent free() of string in .rodata when overriding AuthorizedKeys in a Match block

* sshd(8): Use default shell /bin/sh if $SHELL is ""

* ssh(1): kill proxy command on fatal() (we already killed it on clean exit);

```
 * ssh(1): install a SIGCHLD handler to reap expiried child process;
   bz#1812

 * Support building against openssl-1.0.0a

Portable OpenSSH Bugfixes:

 * Use mandoc as preferred manpage formatter if it is present, followed
   by nroff and groff respectively.

 * sshd(8): Relax permission requirement on btmp logs to allow group
   read/write

 * bz#1840: fix warning when configuring --with-ssl-engine

 * sshd(8): Use correct uid_t/pid_t types instead of int. bz#1817

 * sshd(8): bz#1824: Add Solaris Project support.

 * sshd(8): Check is_selinux_enabled for exact return code since it can
   apparently return -1 under some conditions.

Checksums:
==========

 - SHA1 (openssh-5.7.tar.gz) = 67cb91772a33fb3a004b39bcdb9148218365494c
 - SHA1 (openssh-5.7p1.tar.gz) = 423e27475f06e1055847dfff7f61e1ac632b5372

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 5.6/5.6p1 (2010-08-23)

```
OpenSSH 5.6 was released on 2010-08-23. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
http://www.openssh.com/donations.html

Changes since OpenSSH 5.5
=========================

Features:

 * Added a ControlPersist option to ssh_config(5) that automatically
   starts a background ssh(1) multiplex master when connecting. This
   connection can stay alive indefinitely, or can be set to
   automatically close after a user-specified duration of inactivity.

 * Hostbased authentication may now use certificate host keys. CA keys
   must be specified in a known_hosts file using the @cert-authority
   marker as described in sshd(8).

 * ssh-keygen(1) now supports signing certificate using a CA key that
   has been stored in a PKCS#11 token.

 * ssh(1) will now log the hostname and address that we connected to at
   LogLevel=verbose after authentication is successful to mitigate
   "phishing" attacks by servers with trusted keys that accept
   authentication silently and automatically before presenting fake
   password/passphrase prompts.

   Note that, for such an attack to be successful, the user must have
   disabled StrictHostKeyChecking (enabled by default) or an attacker
   must have access to a trusted host key for the destination server.

 * Expand %h to the hostname in ssh_config Hostname options. While this
   sounds useless, it is actually handy for working with unqualified
   hostnames:

     Host *.*
        Hostname %h
     Host *
        Hostname %h.example.org

 * Allow ssh-keygen(1) to import (-i) and export (-e) of PEM and PKCS#8
   keys in addition to RFC4716 (SSH.COM) encodings via a new -m option
   (bz#1749)

 * sshd(8) will now queue debug messages for bad ownership or
   permissions on the user's keyfiles encountered during authentication
   and will send them after authentication has successfully completed.
```

These messages may be viewed in ssh(1) at LogLevel=debug or higher.

* ssh(1) connection multiplexing now supports remote forwarding with dynamic port allocation and can report the allocated port back to the user:

```
LPORT=`ssh -S muxsocket -R0:localhost:25 -O forward somehost`
```

* sshd(8) now supports indirection in matching of principal names listed in certificates. By default, if a certificate has an embedded principals list then the username on the server must match one of the names in the list for it to be accepted for authentication.

  sshd(8) now has a new AuthorizedPrincipalsFile option to specify a file containing a list of names that may be accepted in place of the username when authorizing a certificate trusted via the sshd_config(5) TrustedCAKeys option. Similarly, authentication using a CA trusted in ~/.ssh/authorized_keys now accepts a principals="name1[,name2,...]" to specify a list of permitted names.

  If either option is absent, the current behaviour of requiring the username to appear in principals continues to apply. These options are useful for role accounts, disjoint account namespaces and "user@realm"-style naming policies in certificates.

* Additional sshd_config(5) options are now valid inside Match blocks:

```
AuthorizedKeysFile
AuthorizedPrincipalsFile
HostbasedUsesNameFromPacketOnly
PermitTunnel
```

* Revised the format of certificate keys. The new format, identified as ssh-{dss,rsa}-cert-v01@openssh.com includes the following changes:

  - Adding a serial number field. This may be specified by the CA at the time of certificate signing.

  - Moving the nonce field to the beginning of the certificate where it can better protect against chosen-prefix attacks on the signature hash (currently infeasible against the SHA1 hash used)

  - Renaming the "constraints" field to "critical options"

  - Addng a new non-critical "extensions" field. The "permit-*" options are now extensions, rather than critical options to permit non-OpenSSH implementation of this key format to degrade gracefully when encountering keys with options they do not recognize.

  The older format is still supported for authentication and may still be used when signing certificates (use "ssh-keygen -t v00 ..."). The v00 format, introduced in OpenSSH 5.4, will be supported for at least one year from this release, after which it will be deprecated and removed.

BugFixes:

* The PKCS#11 code now retries a lookup for a private key if there is no matching key with CKA_SIGN attribute enabled; this fixes fixes MuscleCard support (bz#1736)

* Unbreak strdelim() skipping past quoted strings (bz#1757). For example, the following directive was not parsed correctly:

```
AllowUsers "blah blah" blah
```

* sftp(1): fix swapped args in upload_dir_internal(), breaking recursive upload depth checks and causing verbose printing of transfers to always be turned on (bz#1797)

* Fix a longstanding problem where if you suspend scp(1) at the password/passphrase prompt the terminal mode is not restored.

* Fix a PKCS#11 crash on some smartcards by validating the length returned for C_GetAttributValue (bz#1773)

* sftp(1): fix ls in working directories that contain globbing characters in their pathnames (bz#1655)

* Print warning for missing home directory when ChrootDirectory=none (bz#1564)

* sftp(1): fix a memory leak in do_realpath() error path (bz#1771)

* ssk-keygen(1): Standardise error messages when attempting to open private key files to include "progname: filename: error reason" (bz#1783)

* Replace verbose and overflow-prone Linebuf code with read_keyfile_line() (bz#1565)

* Include the user name on "subsystem request for ..." log messages

* ssh(1) and sshd(8): remove hardcoded limit of 100 permitopen clauses and port forwards per direction (bz#1327)

```
 * sshd(8): ignore stderr output from subsystems to avoid hangs if a
   subsystem or shell initialisation writes to stderr (bz#1750)

 * Skip the initial check for access with an empty password when
   PermitEmptyPasswords=no (bz#1638)

 * sshd(8): fix logspam when key options (from="..." especially) deny
   non-matching keys (bz#1765)

 * ssh-keygen(1): display a more helpful error message when $HOME is
   inaccessible while trying to create .ssh directory (bz#1740)

 * ssh(1): fix hang when terminating a mux slave using ~. (bz#1758)

 * ssh-keygen(1): refuse to generate keys longer than
   OPENSSL_[RD]SA_MAX_MODULUS_BITS, since we would refuse to use
   them anyway (bz#1516)

 * Suppress spurious tty warning when using -O and stdin is not a tty
   (bz#1746)

 * Kill channel when pty allocation requests fail. Fixed stuck client
   if the server refuses pty allocation (bz#1698)

Portable OpenSSH Bugfixes:

 * sshd(8): increase the maximum username length for login recording
   to 512 characters (bz#1579)

 * Initialize the values to be returned from PAM to sane values in
   case the PAM method doesn't write to them. (bz#1795)

 * Let configure find OpenSSL libraries in a lib64 subdirectory.
   (bz#1756)

Checksums:
==========

 - SHA1 (openssh-5.6.tar.gz) = fa5ac394b874d6709031306b6ac5c48399697f7f
 - SHA1 (openssh-5.6p1.tar.gz) = 347dd39c91c3529f41dae63714d452fb95efea1e

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 5.5/5.5p1 (2010-04-16)

```
OpenSSH 5.5 was released on 2010-04-16. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed code
or patches, reported bugs, tested snapshots or donated to the project.
More information on donations may be found at:
http://www.openssh.com/donations.html

This is a bugfix release.

Changes since OpenSSH 5.4
=========================

 * Unbreak sshd_config's AuthorizedKeysFile option for $HOME-relative paths

 * Fix compilation failures on platforms that lack dlopen()

 * Include a language tag when sending a protocol 2 disconnection message.

 * Make logging of certificates used for user authentication more clear and
   consistent between CAs specified using TrustedUserCAKeys and
   authorized_keys

Portable OpenSSH:

 * Allow contrib/ssh-copy-id to fail gracefully when there are no keys in
   the ssh-agent. bz#1723

 * Explicitly link libX11 into contrib/gnome-ssh-askpass2. bz#1725

 * Allow ChrootDirectory to work in SELinux platforms. bz#1726

 * Add configure.ac stanza for Haiku OS. bz#1741

 * Enable utmpx support on FreeBSD where possible. bz#1732

 * Use pkg-config to determine libedit linker flags where possible. bz#1744
```

```
Checksums:
==========

 - SHA1 (openssh-5.5.tar.gz) = 59864a048b09ad1b6e65a74d5d385d8189ab8c74
 - SHA1 (openssh-5.5p1.tar.gz) = 361c6335e74809b26ea096b34062ba8ff6c97cd6

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 5.4/5.4p1 (2010-03-08)

```
OpenSSH 5.4 was released on 2010-03-08. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed code
or patches, reported bugs, tested snapshots or donated to the project.
More information on donations may be found at:
http://www.openssh.com/donations.html

This is a major feature and bugfix release.

Changes since OpenSSH 5.3
=========================

Features:

 * After a transition period of about 10 years, this release disables
   SSH protocol 1 by default. Clients and servers that need to use the
   legacy protocol must explicitly enable it in ssh_config / sshd_config
   or on the command-line.

 * Remove the libsectok/OpenSC-based smartcard code and add support for
   PKCS#11 tokens. This support is automatically enabled on all
   platforms that support dlopen(3) and was inspired by patches written
   by Alon Bar-Lev. Details in the ssh(1) and ssh-add(1) manpages.

 * Add support for certificate authentication of users and hosts using a
   new, minimal OpenSSH certificate format (not X.509). Certificates
   contain a public key, identity information and some validity
   constraints and are signed with a standard SSH public key using
   ssh-keygen(1). CA keys may be marked as trusted in authorized_keys
   or via a TrustedUserCAKeys option in sshd_config(5) (for user
   authentication), or in known_hosts (for host authentication).

   Documentation for certificate support may be found in ssh-keygen(1),
   sshd(8) and ssh(1) and a description of the protocol extensions in
   PROTOCOL.certkeys.

 * Added a 'netcat mode' to ssh(1): "ssh -W host:port ..." This connects
   stdio on the client to a single port forward on the server. This
   allows, for example, using ssh as a ProxyCommand to route connections
   via intermediate servers. bz#1618

 * Add the ability to revoke keys in sshd(8) and ssh(1). User keys may
   be revoked using a new sshd_config(5) option "RevokedKeys". Host keys
   are revoked through known_hosts (details in the sshd(8) man page).
   Revoked keys cannot be used for user or host authentication and will
   trigger a warning if used.

 * Rewrite the ssh(1) multiplexing support to support non-blocking
   operation of the mux master, improve the resilience of the master to
   malformed messages sent to it by the slave and add support for
   requesting port- forwardings via the multiplex protocol. The new
   stdio-to-local forward mode ("ssh -W host:port ...") is also
   supported. The revised multiplexing protocol is documented in the
   file PROTOCOL.mux in the source distribution.

 * Add a 'read-only' mode to sftp-server(8) that disables open in write
   mode and all other fs-modifying protocol methods. bz#430

 * Allow setting an explicit umask on the sftp-server(8) commandline to
   override whatever default the user has. bz#1229

 * Many improvements to the sftp(1) client, many of which were
   implemented by Carlos Silva through the Google Summer of Code
   program:
   - Support the "-h" (human-readable units) flag for ls
   - Implement tab-completion of commands, local and remote filenames
   - Support most of scp(1)'s commandline arguments in sftp(1), as a
     first step towards making sftp(1) a drop-in replacement for scp(1).
     Note that the rarely-used "-P sftp_server_path" option has been
     moved to "-D sftp_server_path" to make way for "-P port" to match
     scp(1).
   - Add recursive transfer support for get/put and on the commandline
```

```
  * New RSA keys will be generated with a public exponent of RSA_F4 ==
    (2**16)+1 == 65537 instead of the previous value 35.

  * Passphrase-protected SSH protocol 2 private keys are now protected
    with AES-128 instead of 3DES. This applied to newly-generated keys
    as well as keys that are reencrypted (e.g. by changing their
    passphrase).

Bugfixes:

  * Hold authentication debug messages until after successful
    authentication. Fixes a minor information leak of environment
    variables specified in authorized_keys if an attacker happens to
    know the public key in use.
  * When using ChrootDirectory, make sure we test for the existence of
    the user's shell inside the chroot and not outside (bz#1679)
  * Cache user and group name lookups in sftp-server using
    user_from_[ug]id(3) to improve performance on hosts where these
    operations are slow (e.g. NIS or LDAP). bz#1495
  * Fix problem that prevented passphrase reading from being interrupted
    in some circumstances; bz#1590
  * Ignore and log any Protocol 1 keys where the claimed size is not
    equal to the actual size.
  * Make HostBased authentication work with a ProxyCommand. bz#1569
  * Avoid run-time failures when specifying hostkeys via a relative
    path by prepending the current working directory in these cases.
    bz#1290
  * Do not prompt for a passphrase if we fail to open a keyfile, and log
    the reason why the open failed to debug. bz#1693
  * Document that the PubkeyAuthentication directive is allowed in a
    sshd_config(5) Match block. bz#1577
  * When converting keys, truncate key comments at 72 chars as per
    RFC4716. bz#1630
  * Do not allow logins if /etc/nologin exists but is not readable by the
    user logging in.
  * Output a debug log if sshd(8) can't open an existing authorized_keys.
    bz#1694
  * Quell tc[gs]etattr warnings when forcing a tty (ssh -tt), since we
    usually don't actually have a tty to read/set; bz#1686
  * Prevent sftp from crashing when given a "-" without a command.
    Also, allow whitespace to follow a "-". bz#1691
  * After sshd receives a SIGHUP, ignore subsequent HUPs while sshd
    re-execs itself. Prevents two HUPs in quick succession from resulting
    in sshd dying. bz#1692
  * Clarify in sshd_config(5) that StrictModes does not apply to
    ChrootDirectory. Permissions and ownership are always checked when
    chrooting. bz#1532
  * Set close-on-exec on various descriptors so they don't get leaked to
    child processes. bz#1643
  * Fix very rare race condition in x11/agent channel allocation: don't
    read after the end of the select read/write fdset and make sure a
    reused FD is not touched before the pre-handlers are called.
  * Fix incorrect exit status when multiplexing and channel ID 0 is
    recycled. bz#1570
  * Fail with an error when an attempt is made to connect to a server
    with ForceCommand=internal-sftp with a shell session (i.e. not a
    subsystem session). Avoids stuck client when attempting to ssh to
    such a service. bz#1606:
  * Warn but do not fail if stat()ing the subsystem binary fails. This
    helps with chrootdirectory+forcecommand=sftp-server and restricted
    shells. bz #1599
  * Change "Connecting to host..." message to "Connected to host."
    and delay it until after the sftp protocol connection has been
    established. Avoids confusing sequence of messages when the
    underlying ssh connection experiences problems. bz#1588
  * Use the HostKeyAlias rather than the hostname specified on the
    commandline when prompting for passwords. bz#1039
  * Correct off-by-one in percent_expand(): we would fatal() when trying
    to expand EXPAND_MAX_KEYS, allowing only EXPAND_MAX_KEYS-1 to
    actually work. Note that nothing in OpenSSH actually uses close to
    this limit at present. bz#1607
  * Fix passing of empty options from scp(1) and sftp(1) to the
    underlying ssh(1). Also add support for the stop option "--".
  * Fix an incorrect magic number and typo in PROTOCOL; bz#1688
  * Don't escape backslashes when displaying the SSH2 banner. bz#1533
  * Don't unnecessarily dup() the in and out fds for sftp-server. bz#1566
  * Force use of the correct hash function for random-art signature
    display as it was inheriting the wrong one when bubblebabble
    signatures were activated. bz#1611
  * Do not fall back to adding keys without constraints (ssh-add -c /
    -t ...) when the agent refuses the constrained add request. bz#1612
  * Fix a race condition in ssh-agent that could result in a wedged or
    spinning agent. bz#1633
  * Flush stdio before exec() to ensure that everying (motd
    in particular) has made it out before the streams go away. bz#1596
  * Set FD_CLOEXEC on in/out sockets in sshd(8). bz#1706

Portable OpenSSH Bugfixes:

  * Use system's kerberos principal name on AIX if it's available.
    bz#1583
  * Disable OOM-killing of the listening sshd on Linux. bz#1470
  * Use pkg-config for opensc config if it's available. bz#1160
  * Unbreak Redhat spec to allow building without askpass. bz#1677
  * If PidFile is set in sshd_config, use it in SMF init file. bz#1628
  * Print error and usage() when ssh-rand-helper is passed command-
    line arguments as none are supported. bz#1568
```

```
 * Add missing setsockopt() to set IPV6_V6ONLY for local forwarding
   with GatwayPorts=yes. bz#1648
 * Make GNOME 2 askpass dialog desktop-modal. bz#1645
 * If SELinux is enabled set the security context to "sftpd_t" before
   running the internal sftp server. bz#1637
 * Correctly check libselinux for necessary SELinux functions; bz#1713
 * Unbreak builds on Redhat using the supplied openssh.spec; bz#1731
 * Fix incorrect privilege dropping order on AIX that prevented
   chroot operation; bz#1567
 * Call aix_setauthdb/aix_restoredb at the correct times on AIX to
   prevent authentication failure; bz#1710

Checksums:
==========

 - SHA1 (openssh-5.4.tar.gz) = 1776832d902f7b4c7863afd41a5ec7a14efe95d6
 - SHA1 (openssh-5.4p1.tar.gz) = 2a3042372f08afb1415ceaec8178213276a36302

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 5.3/5.3p1 (2009-10-01)

```
OpenSSH 5.3 was released on 2009-10-01. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

This release marks the 10th anniversary of the OpenSSH project.
We would like to thank the OpenSSH community for their support,
especially those who will continue to contribute code or patches,
report bugs, test snapshots or donate to the project during the
next 10 years.  More information on donations may be found at:
http://www.openssh.com/donations.html

This is a bugfix release, no new features have been added.

Changes since OpenSSH 5.2
=========================

General Bugfixes:

 * Do not limit home directory paths to 256 characters. bz#1615

 * Several minor documentation and correctness fixes.

Portable OpenSSH Bugfixes:

 * This release removes for support for very old versions of Cygwin and
   for Windows 95/98/ME

 * Move the deletion of PAM credentials on logout to after the session
   close. bz#1534

 * Make PrintLastLog work on AIX. bz#1595

 * Avoid compile errors on FreeBSD from conflicts in glob.h. bz#1634

 * Delay dropping of root privileges on AIX so chroot and pam_open_session
   work correctly. bz#1249 and bz#1567

 * Increase client IO buffer on Cygwin to 64K, realising a significant
   performance improvement.

 * Roll back bz#1241 (better handling for expired passwords on Tru64).
   The change broke password logins on some configurations.

 * Accept ENOSYS as a fallback error when attempting atomic
   rename(). bz#1535

 * Fix passing of variables to recursive make(1) invocations on Solaris.
   bz#1505

 * Skip the tcgetattr call on the pty master on Solaris, since it never
   succeeds and can hang if large amounts of data is sent to the slave
   (eg a copy-paste). bz#1528

 * Fix detection of krb5-config. bz#1639

 * Fix test for server-assigned remote forwarding port for non-root users.
   bz#1578

 * Fix detection of libresolv on OSX 10.6.

Checksums:
==========

 - SHA1 (openssh-5.3.tar.gz) = f1b9a280565e916c1f84fd4d944313ec926242a2
```

```
 - SHA1 (openssh-5.3p1.tar.gz) = d411fde2584ef6022187f565360b2c63a05602b5

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 5.2/5.2p1 (2009-02-23)

```
OpenSSH 5.2 was released on 2009-02-23. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

We have also recently completed another Internet SSH usage scan, the
results of which may be found at http://www.openssh.com/usage.html

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots or donated to the
project. More information on donations may be found at:
http://www.openssh.com/donations.html

The focus of this release has been on bugfixes as the previous
openssh-5.1 release introduced many new features and made some
invasive changes.

Changes since OpenSSH 5.1
=========================

Security:

 * This release changes the default cipher order to prefer the AES CTR
   modes and the revised "arcfour256" mode to CBC mode ciphers that are
   susceptible to CPNI-957037 "Plaintext Recovery Attack Against SSH".

 * This release also adds countermeasures to mitigate CPNI-957037-style
   attacks against the SSH protocol's use of CBC-mode ciphers. Upon
   detection of an invalid packet length or Message Authentication
   Code, ssh/sshd will continue reading up to the maximum supported
   packet length rather than immediately terminating the connection.
   This eliminates most of the known differences in behaviour that
   leaked information about the plaintext of injected data which formed
   the basis of this attack. We believe that these attacks are rendered
   infeasible by these changes.

New features:

 * Added a -y option to ssh(1) to force logging to syslog rather than
   stderr, which is useful when running daemonised (ssh -f)

 * The sshd_config(5) ForceCommand directive now accepts commandline
   arguments for the internal-sftp server.

 * The ssh(1) ~C escape commandline now support runtime creation of
   dynamic (-D) port forwards.

 * Support the SOCKS4A protocol in ssh(1) dynamic (-D) forwards.
   (bz#1482)

 * Support remote port forwarding with a listen port of '0'. This
   informs the server that it should dynamically allocate a listen
   port and report it back to the client. (bz#1003)

 * sshd(8) now supports setting PermitEmptyPasswords and
   AllowAgentForwarding in Match blocks

Bug and documentation fixes

 * Repair a ssh(1) crash introduced in openssh-5.1 when the client is
   sent a zero-length banner (bz#1496)

 * Due to interoperability problems with certain
   broken SSH implementations, the eow@openssh.com and
   no-more-sessions@openssh.com protocol extensions are now only sent
   to peers that identify themselves as OpenSSH.

 * Make ssh(1) send the correct channel number for
   SSH2_MSG_CHANNEL_SUCCESS and SSH2_MSG_CHANNEL_FAILURE messages to
   avoid triggering 'Non-public channel' error messages on sshd(8) in
   openssh-5.1.

 * Avoid printing 'Non-public channel' warnings in sshd(8), since the
   ssh(1) has sent incorrect channel numbers since ~2004 (this reverts
   a behaviour introduced in openssh-5.1).

 * Avoid double-free in ssh(1) ~C escape -L handler (bz#1539)

 * Correct fail-on-error behaviour in sftp(1) batchmode for remote
```

stat operations. (bz#1541)

* Disable nonfunctional ssh(1) ~C escape handler in multiplex slave connections. (bz#1543)

* Avoid hang in ssh(1) when attempting to connect to a server that has MaxSessions=0 set.

* Multiple fixes to sshd(8) configuration test (-T) mode

* Several core and portable OpenSSH bugs fixed: 1380, 1412, 1418, 1419, 1421, 1490, 1491, 1492, 1514, 1515, 1518, 1520, 1538, 1540

* Many manual page improvements.

Checksums:
==========

- SHA1 (openssh-5.2.tar.gz) = 260074ed466e95f054ac05a4406f613d08575217
- SHA1 (openssh-5.2p1.tar.gz) = 8273a0237db98179fbdc412207ff8eb14ff3d6de

Reporting Bugs:
===============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt, Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and Ben Lindstrom.

OpenSSH 5.1/5.1p1 (2008-07-22)

OpenSSH 5.1 was released on 2008-07-22. It is available from the mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0 implementation and includes sftp client and server support.

We have also recently completed another Internet SSH usage scan, the results of which may be found at http://www.openssh.com/usage.html

Once again, we would like to thank the OpenSSH community for their continued support of the project, especially those who contributed code or patches, reported bugs, tested snapshots or donated to the project. More information on donations may be found at:
http://www.openssh.com/donations.html

Changes since OpenSSH 5.0
=========================

Security:

* sshd(8): Avoid X11 man-in-the-middle attack on HP/UX (and possibly other platforms) when X11UseLocalhost=no

  When attempting to bind(2) to a port that has previously been bound with SO_REUSEADDR set, most operating systems check that either the effective user-id matches the previous bind (common on BSD-derived systems) or that the bind addresses do not overlap (Linux and Solaris).

  Some operating systems, such as HP/UX, do not perform these checks and are vulnerable to an X11 man-in-the-middle attack when the sshd_config(5) option X11UseLocalhost has been set to "no" - an attacker may establish a more-specific bind, which will be used in preference to sshd's wildcard listener.

  Modern BSD operating systems, Linux, OS X and Solaris implement the above checks and are not vulnerable to this attack, nor are systems where the X11UseLocalhost has been left at the default value of "yes".

  Portable OpenSSH 5.1 avoids this problem for all operating systems by not setting SO_REUSEADDR when X11UseLocalhost is set to no.

  This vulnerability was reported by sway2004009 AT hotmail.com.

New features:

* Introduce experimental SSH Fingerprint ASCII Visualisation to ssh(1) and ssh-keygen(1). Visual fingerprinnt display is controlled by a new ssh_config(5) option "VisualHostKey". The intent is to render SSH host keys in a visual form that is amenable to easy recall and rejection of changed host keys. This technique inspired by the graphical hash visualisation schemes known as "random art[*]", and by Dan Kaminsky's musings at 23C3 in Berlin.

  Fingerprint visualisation in is currently disabled by default, as the algorithm used to generate the random art is still subject to change.

  [*] "Hash Visualization: a New Technique to improve Real-World Security", Perrig A. and Song D., 1999, International Workshop on Cryptographic Techniques and E-Commerce (CrypTEC '99)
  http://sparrow.ece.cmu.edu/~adrian/projects/validation/validation.pdf

* sshd_config(5) now supports CIDR address/masklen matching in "Match address" blocks, with a fallback to classic wildcard matching. For example:
  ```
  Match address 192.0.2.0/24,3ffe:ffff::/32,!10.*
      PasswordAuthentication yes
  ```

* sshd(8) now supports CIDR matching in ~/.ssh/authorized_keys from="..." restrictions, also with a fallback to classic wildcard matching.

* Added an extended test mode (-T) to sshd(8) to request that it write its effective configuration to stdout and exit. Extended test mode also supports the specification of connection parameters (username, source address and hostname) to test the application of sshd_config(5) Match rules.

* ssh(1) now prints the number of bytes transferred and the overall connection throughput for SSH protocol 2 sessions when in verbose mode (previously these statistics were displayed for protocol 1 connections only).

* sftp-server(8) now supports extension methods statvfs@openssh.com and fstatvfs@openssh.com that implement statvfs(2)-like operations. (bz#1399)

* sftp(1) now has a "df" command to the sftp client that uses the statvfs@openssh.com to produce a df(1)-like display of filesystem space and inode utilisation (requires statvfs@openssh.com support on the server)

* Added a MaxSessions option to sshd_config(5) to allow control of the number of multiplexed sessions supported over a single TCP connection. This allows increasing the number of allowed sessions above the previous default of 10, disabling connection multiplexing (MaxSessions=1) or disallowing login/shell/subsystem sessions entirely (MaxSessions=0).

* Added a no-more-sessions@openssh.com global request extension that is sent from ssh(1) to sshd(8) when the client knows that it will never request another session (i.e. when session multiplexing is disabled). This allows a server to disallow further session requests and terminate the session in cases where the client has been hijacked.

* ssh-keygen(1) now supports the use of the -l option in combination with -F to search for a host in ~/.ssh/known_hosts and display its fingerprint.

* ssh-keyscan(1) now defaults to "rsa" (protocol 2) keys, instead of "rsa1".

* Added an AllowAgentForwarding option to sshd_config(8) to control whether authentication agent forwarding is permitted. Note that this is a loose control, as a client may install their own unofficial forwarder.

* ssh(1) and sshd(8): avoid unnecessary malloc/copy/free when receiving network data, resulting in a ~10% speedup

* ssh(1) and sshd(8) will now try additional addresses when connecting to a port forward destination whose DNS name resolves to more than one address. The previous behaviour was to try the only first address and give up if that failed. (bz#383)

* ssh(1) and sshd(8) now support signalling that channels are half-closed for writing, through a channel protocol extension notification "eow@openssh.com". This allows propagation of closed file descriptors, so that commands such as:
  ```
  "ssh -2 localhost od /bin/ls | true"
  ```
  do not send unnecessary data over the wire. (bz#85)

* sshd(8): increased the default size of ssh protocol 1 ephemeral keys from 768 to 1024 bits.

* When ssh(1) has been requested to fork after authentication ("ssh -f") with ExitOnForwardFailure enabled, delay the fork until after replies for any -R forwards have been seen. Allows for robust detection of -R forward failure when using -f. (bz#92)

* "Match group" blocks in sshd_config(5) now support negation of groups. E.g. "Match group staff,!guests" (bz#1315)

* sftp(1) and sftp-server(8) now allow chmod-like operations to set set[ug]id/sticky bits. (bz#1310)

* The MaxAuthTries option is now permitted in sshd_config(5) match blocks.

* Multiplexed ssh(1) sessions now support a subset of the ~ escapes that are available to a primary connection. (bz#1331)

* ssh(1) connection multiplexing will now fall back to creating a new connection in most error cases. (bz#1439 bz#1329)

* Added some basic interoperability tests against Twisted Conch.

* Documented OpenSSH's extensions to and deviations from the published SSH protocols (the PROTOCOL file in the distribution)

```
   * Documented OpenSSH's ssh-agent protocol (PROTOCOL.agent).

  Bug and documentation fixes

   * Make ssh(1) deal more gracefully with channel requests that fail.
     Previously it would optimistically assume that requests would always
     succeed, which could cause hangs if they did not (e.g. when the
     server runs out of file descriptors). (bz#1384)

   * ssh(1) now reports multiplexing errors via the multiplex slave's
     stderr where possible (subject to LogLevel in the mux master).

   * ssh(1) and sshd(8) now send terminate protocol banners with CR+LF for
     protocol 2 to comply with RFC 4253. Previously they were terminated
     with CR alone. Protocol 1 banners remain CR terminated. (bz#1443)

   * Merged duplicate authentication file checks in sshd(8) and refuse to
     read authorised_keys and .shosts from non-regular files. (bz#1438)

   * Ensure that sshd(8)'s umask disallows at least group and world write,
     even if a more permissive one has been inherited. (bz#1433)

   * Suppress the warning message from sshd(8) when changing to a
     non-existent user home directory after chrooting. (bz#1461)

   * Mention that scp(1) follows symlinks when performing recursive
     copies. (bz#1466)

   * Prevent sshd(8) from erroneously applying public key restrictions
     leaned from ~/.ssh/authorized_keys to other authentication methods
     when public key authentication subsequently fails. (bz#1472)

   * Fix protocol keepalive timeouts - in some cases, keepalive packets
     were being sent, but the connection was not being closed when the
     limit for missing replies was exceeded. (bz#1465)

   * Fix ssh(1) sending invalid TTY modes when a TTY was forced (ssh -tt)
     but stdin was not a TTY. (bz#1199)

   * ssh(1) will now exit with a non-zero exit status if
     ExitOnForwardFailure was set and forwardings were disabled due to a
     failed host key check.

   * Fix MaxAuthTries tests to disallow a free authentication try to
     clients that skipped the protocol 2 "none" authentication method.
     (part of bz#1432)

   * Make keepalive timeouts apply while synchronously waiting
     for a packet, particularly during key renegotiation. (bz#1363)

   * sshd(8) has been audited to eliminate fd leaks and calls to fatal()
     in conditions of file descriptor exhaustion.

  Portable OpenSSH-specific bugfixes

   * Avoid a sshd(8) hang-on-exit on Solaris caused by depending on the
     success of isatty() on a PTY master (undefined behaviour). Probably
     affected other platforms too. (bz#1463)

   * Fixed test for locked accounts on HP/UX with shadowed
     passwords disabled. (bz#1083)

   * Disable poll() fallback in atomiciov for Tru64. readv
     doesn't seem to be a comparable object there, which lead to
     compilation errors. (bz#1386)

   * Fall back to racy rename if link returns EXDEV. (bz#1447)

   * Explicitly handle EWOULDBLOCK wherever we handle EAGAIN, on
     some platforms (HP nonstop) it is a distinct errno. (bz#1467)

   * Avoid NULL dereferences in ancient sigaction replacement
     code. (bz#1240)

   * Avoid linking against libgssapi, which despite its name
     doesn't seem to implement all of GSSAPI. (bz#1276)

   * Use explicit noreturn attribute instead of __dead, fixing
     compilation problems on Interix. (bz#1112)

   * Added support password expiry on Tru64 SIA systems. (bz#1241)

   * Fixed an UMAC alignment problem that manifested on Itanium
     platforms. (bz#1462)

   * The sftp-server(8) manual now describes the requirements for
     transfer logging in chroot environments. (bz#1488)

   * Strip trailing dot from hostnames when the sshd_config(5)
     HostbasedUsesNameFromPacketOnly option is set. (bz#1200)

  Checksums:
  ==========

   - SHA1 (openssh-5.1.tar.gz) = 1e5b43844ed015e4fbbbe25cfad6f5377c60e759
   - SHA1 (openssh-5.1p1.tar.gz) = 877ea5b283060fe0160e376ea645e8e168047ff5

  Reporting Bugs:
```

```
==============

- Please read http://www.openssh.com/report.html
  Security bugs should be reported directly to openssh@openssh.com

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 5.0/5.0p1 (2008-04-03)

```
OpenSSH 5.0 was released on 2008-04-03. It is available from the
mirrors listed at https://www.openssh.com/.
We apologise for any inconvenience resulting from this release
being made so shortly after 4.9. Unfortunately we only learned of
the below security issue from the public CVE report. The Debian
OpenSSH maintainers responsible for handling the initial report of
this bug failed to report it via either the private OpenSSH security
contact list (openssh@openssh.com) or the portable OpenSSH Bugzilla
(http://bugzilla.mindrot.org/).
We ask anyone wishing to report security bugs in OpenSSH to please use
the openssh@openssh.com contact and to practice responsible disclosure.

OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots and purchased
T-shirts or posters.

T-shirt, poster and CD sales directly support the project. Pictures
and more information can be found at:
        http://www.openbsd.org/tshirts.html and
        http://www.openbsd.org/orders.html

For international orders use http://https.openbsd.org/cgi-bin/order
and for European orders, use http://https.openbsd.org/cgi-bin/order.eu

Changes since OpenSSH 4.9:
============================

Security:

 * CVE-2008-1483: Avoid possible hijacking of X11-forwarded connections
   by refusing to listen on a port unless all address families bind
   successfully.

Checksums:
==========

 - SHA1 (openssh-5.0.tar.gz) = 729fb3168edf6a68408223b5ed82e59d13b57c47
 - SHA1 (openssh-5.0p1.tar.gz) = 121cea3a730c0b0353334b6f46f438de30ab4928

Reporting Bugs:
===============

- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 4.9/4.9p1 (2008-03-31)

```
OpenSSH 4.9 was released on 2008-03-31. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots and purchased
T-shirts or posters.

T-shirt, poster and CD sales directly support the project. Pictures
and more information can be found at:
        http://www.openbsd.org/tshirts.html and
        http://www.openbsd.org/orders.html

For international orders use http://https.openbsd.org/cgi-bin/order
and for European orders, use http://https.openbsd.org/cgi-bin/order.eu

Note that OpenSSH 4.8 was an OpenBSD-only release shipped with the
OpenBSD 4.3 CD.

Changes since OpenSSH 4.7:
============================

Security:
```

* Disable execution of ~/.ssh/rc for sessions where a command has been forced by the sshd_config ForceCommand directive. Users who had write access to this file could use it to execute abritrary commands. This behaviour was documented, but was an unsafe default and an extra hassle for administrators.

New features:

* Added chroot(2) support for sshd(8), controlled by a new option "ChrootDirectory". Please refer to sshd_config(5) for details, and please use this feature carefully. (bz#177 bz#1352)
* Linked sftp-server(8) into sshd(8). The internal sftp server is used when the command "internal-sftp" is specified in a Subsystem or ForceCommand declaration. When used with ChrootDirectory, the internal sftp server requires no special configuration of files inside the chroot environment. Please refer to sshd_config(5) for more information.
* Added a "no-user-rc" option for authorized_keys to disable execution of ~/.ssh/rc
* Added a protocol extension method "posix-rename@openssh.com" for sftp-server(8) to perform POSIX atomic rename() operations. (bz#1400)
* Removed the fixed limit of 100 file handles in sftp-server(8). The server will now dynamically allocate handles up to the number of available file descriptors. (bz#1397)
* ssh(8) will now skip generation of SSH protocol 1 ephemeral server keys when in inetd mode and protocol 2 connections are negotiated. This speeds up protocol 2 connections to inetd-mode servers that also allow Protocol 1 (bz#440)
* Accept the PermitRootLogin directive in a sshd_config(5) Match block. Allows for, e.g. permitting root only from the local network.
* Reworked sftp(1) argument splitting and escaping to be more internally consistent (i.e. between sftp commands) and more consistent with sh(1). Please note that this will change the interpretation of some quoted strings, especially those with embedded backslash escape sequences. (bz#778)
* Support "Banner=none" in sshd_config(5) to disable sending of a pre-login banner (e.g. in a Match block).
* ssh(1) ProxyCommands are now executed with $SHELL rather than /bin/sh.
* ssh(1)'s ConnectTimeout option is now applied to both the TCP connection and the SSH banner exchange (previously it just covered the TCP connection). This allows callers of ssh(1) to better detect and deal with stuck servers that accept a TCP connection but don't progress the protocol, and also makes ConnectTimeout useful for connections via a ProxyCommand.
* Many new regression tests, including interop tests against PuTTY's plink.
* Support BSM auditing on Mac OS X

The following bugs have been fixed in this release:

- scp(1) incorrectly reported "stalled" on slow copies. (bz#799)
- scp(1) date underflow for timestamps before epoch. (bz#828)
- scp(1) and sftp(1) progressmeter type fixes. (bz#842)
- SSH2_MSG_UNIMPLEMENTED packets did not correctly reset the client keepalive logic, causing disconnections on servers that did not explicitly implement "keepalive@openssh.com". (bz#1307)
- ssh(1) used the obsolete SIG DNS RRtype for host keys in DNS, instead of the current standard RRSIG. (bz#1317)
- Extract magic buffer size constants in scp(1) to #defines. (bz#1333)
- Correctly drain ACKs when a sftp(1) upload write fails midway, avoids a fatal() exit from what should be a recoverable condition. (bz#1354)
- Avoid pointer arithmetic and strict aliasing warnings. (bz#1355)
- Fixed packet size advertisements. Previously TCP and agent forwarding incorrectly advertised the channel window size as the packet size, causing fatal errors under some conditions. (bz#1360)
- Document KbdInteractiveAuthentication in sshd_config(5). (bz#1367)
- Fixed sftp(1) file handle leak on download when the local file could not be opened. (bz#1375)
- Fixed ssh-keygen(1) selective host key hashing (i.e. "ssh-keygen -HF hostname") to not include any IP address in the data to be hashed. (bz#1376)
- Fix clobbering of struct passwd from multiple getpwid calls, resulting in fatal errors inside tilde_expand_filename. (bz#1377)
- Fix parsing of port-forwarding specifications to correctly detect errors in either specified port number. (bz#1378)
- Fix memory leak in ssh(1) ~ escape commandline handling. (bz#1379)
- Make ssh(1) skip listening on the IPv6 wildcard address when a binding address of 0.0.0.0 is used against an old SSH server that does not support the RFC4254 syntax for wildcard bind addresses. (bz#1381)
- Remove extra backslashes in the RB_PROTOTYPE macro definition. (bz#1385)
- Support ssh(1) RekeyLimits up to the maximum allowed by the protocol: 2**32-1. (bz#1390)
- Enable IPV6_V6ONLY socket option on sshd(8) listen socket, as is already done for X11/TCP forwarding sockets. (bz#1392)
- Fix FD leak that could hang a ssh(1) connection multiplexing master. (bz#1398)
- Improve error messages when hostname resolution fails due to a system error. (bz#1417)
- Make ssh(1) -q option documentation consistent with reality. (bz#1427 bz#1429)

```
Portable OpenSSH bugs fixed:

   - Fixed sshd(8) PAM support not calling pam_session_close(), or
     failing to call it with root privileges. (bz#926)
   - Made sshd(8) use configuration-specified SyslogFacility for
     hosts_access(3) messages for denied connections. (bz#1042)
   - Implement getgrouplist(3) for AIX, enabling NSS LDAP to work on
     this platform. (bz#1081)
   - Fix compilation errors on AIX due to misdefinition of LLONG_MAX.
     (bz#1347)
   - Force use of local glob(3) implementation on Mac OS X and FreeBSD,
     as the platform versions lack features or have unexpected
     behaviour. (bz#1407)
   - Reduce stdout/stderr noise from ssh-copy-id. (bz#1431)
   - Fix activation of OpenSSL engine support when requested in
     configure. (bz#1437)

Checksums:
==========

 - SHA1 (openssh-4.9.tar.gz) = fa7d1b3dcb093bd0dfc643b33b1a57a26f459373
 - SHA1 (openssh-4.9p1.tar.gz) = 91575878883065bd777f82b47e0d481ac69ee7fe

Reporting Bugs:
===============

- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 4.8/4.8p1 (2008-03-31)

```
OpenSSH 4.8 was an OpenBSD-only release, included on the OpenBSD 4.3
CD only.

Changes since OpenSSH 4.8:
============================

New features:

  * Added chroot(2) support for sshd(8), controlled by a new option
    "ChrootDirectory". Please refer to sshd_config(5) for details, and
    please use this feature carefully. (bz#177 bz#1352)
  * Linked sftp-server(8) into sshd(8). The internal sftp server is
    used when the command "internal-sftp" is specified in a Subsystem
    or ForceCommand declaration. When used with ChrootDirectory, the
    internal sftp server requires no special configuration of files
    inside the chroot environment. Please refer to sshd_config(5) for
    more information.
  * Added a protocol extension method "posix-rename@openssh.com" for
    sftp-server(8) to perform POSIX atomic rename() operations.
    (bz#1400)
  * Removed the fixed limit of 100 file handles in sftp-server(8). The
    server will now dynamically allocate handles up to the number of
    available file descriptors. (bz#1397)
  * ssh(8) will now skip generation of SSH protocol 1 ephemeral server
    keys when in inetd mode and protocol 2 connections are negotiated.
    This speeds up protocol 2 connections to inetd-mode servers that
    also allow Protocol 1 (bz#440)
  * Accept the PermitRootLogin directive in a sshd_config(5) Match
    block. Allows for, e.g. permitting root only from the local
    network.
  * Reworked sftp(1) argument splitting and escaping to be more
    internally consistent (i.e. between sftp commands) and more
    consistent with sh(1). Please note that this will change the
    interpretation of some quoted strings, especially those with
    embedded backslash escape sequences. (bz#778)
  * Support "Banner=none" in sshd_config(5) to disable sending of a
    pre-login banner (e.g. in a Match block).
  * ssh(1) ProxyCommands are now executed with $SHELL rather than
    /bin/sh.
  * ssh(1)'s ConnectTimeout option is now applied to both the TCP
    connection and the SSH banner exchange (previously it just covered
    the TCP connection). This allows callers of ssh(1) to better detect
    and deal with stuck servers that accept a TCP connection but don't
    progress the protocol, and also makes ConnectTimeout useful for
    connections via a ProxyCommand.
  * Many new regression tests, including interop tests against PuTTY's
    plink.
  * Support BSM auditing on Mac OS X

The following bugs have been fixed in this release:

   - scp(1) incorrectly reported "stalled" on slow copies. (bz#799)
   - scp(1) date underflow for timestamps before epoch. (bz#828)
   - scp(1) and sftp(1) progressmeter type fixes. (bz#842)
   - SSH2_MSG_UNIMPLEMENTED packets did not correctly reset the client
     keepalive logic, causing disconnections on servers that did not
     explicitly implement "keepalive@openssh.com". (bz#1307)
   - ssh(1) used the obsolete SIG DNS RRtype for host keys in DNS,
     instead of the current standard RRSIG. (bz#1317)
   - Extract magic buffer size constants in scp(1) to #defines.
```

```
      (bz#1333)
    - Correctly drain ACKs when a sftp(1) upload write fails midway,
      avoids a fatal() exit from what should be a recoverable condition.
      (bz#1354)
    - Avoid pointer arithmetic and strict aliasing warnings. (bz#1355)
    - Fixed packet size advertisements. Previously TCP and agent
      forwarding incorrectly advertised the channel window size as the
      packet size, causing fatal errors under some conditions. (bz#1360)
    - Document KbdInteractiveAuthentication in sshd_config(5). (bz#1367)
    - Fixed sftp(1) file handle leak on download when the local file
      could not be opened. (bz#1375)
    - Fixed ssh-keygen(1) selective host key hashing (i.e.
      "ssh-keygen -HF hostname") to not include any IP address in the
      data to be hashed. (bz#1376)
    - Fix clobbering of struct passwd from multiple getpwid calls,
      resulting in fatal errors inside tilde_expand_filename. (bz#1377)
    - Fix parsing of port-forwarding specifications to correctly
      detect errors in either specified port number. (bz#1378)
    - Fix memory leak in ssh(1) ~ escape commandline handling. (bz#1379)
    - Make ssh(1) skip listening on the IPv6 wildcard address when a
      binding address of 0.0.0.0 is used against an old SSH server that
      does not support the RFC4254 syntax for wildcard bind addresses.
      (bz#1381)
    - Remove extra backslashes in the RB_PROTOTYPE macro definition.
      (bz#1385)
    - Support ssh(1) RekeyLimits up to the maximum allowed by the
      protocol: 2**32-1. (bz#1390)
    - Enable IPV6_V6ONLY socket option on sshd(8) listen socket, as is
      already done for X11/TCP forwarding sockets. (bz#1392)
    - Fix FD leak that could hang a ssh(1) connection multiplexing
      master. (bz#1398)
    - Improve error messages when hostname resolution fails due to a
      system error. (bz#1417)
    - Make ssh(1) -q option documentation consistent with reality.
      (bz#1427 bz#1429)

  Portable OpenSSH bugs fixed:

    - Fixed sshd(8) PAM support not calling pam_session_close(), or
      failing to call it with root privileges. (bz#926)
    - Made sshd(8) use configuration-specified SyslogFacility for
      hosts_access(3) messages for denied connections. (bz#1042)
    - Implement getgrouplist(3) for AIX, enabling NSS LDAP to work on
      this platform. (bz#1081)
    - Fix compilation errors on AIX due to misdefinition of LLONG_MAX.
      (bz#1347)
    - Force use of local glob(3) implementation on Mac OS X and FreeBSD,
      as the platform versions lack features or have unexpected
      behaviour. (bz#1407)
    - Reduce stdout/stderr noise from ssh-copy-id. (bz#1431)
    - Fix activation of OpenSSL engine support when requested in
      configure. (bz#1437)

  Reporting Bugs:
  ===============

  - please read http://www.openssh.com/report.html
    and http://bugzilla.mindrot.org/

  OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
  Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
  Ben Lindstrom.
```

OpenSSH 4.7/4.7p1 (2007-09-05)

```
OpenSSH 4.7 was released on 2007-09-05. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots and purchased
T-shirts or posters.

T-shirt, poster and CD sales directly support the project. Pictures
and more information can be found at:
        http://www.openbsd.org/tshirts.html and
        http://www.openbsd.org/orders.html

For international orders use http://https.openbsd.org/cgi-bin/order
and for European orders, use http://https.openbsd.org/cgi-bin/order.eu

Changes since OpenSSH 4.6:
==========================

Security bugs resolved in this release:

 * Prevent ssh(1) from using a trusted X11 cookie if creation of an
   untrusted cookie fails; found and fixed by Jan Pechanec.

Other changes, new functionality and fixes in this release:

 * sshd(8) in new installations defaults to SSH Protocol 2 only.
   Existing installations are unchanged.
```

```
  * The SSH channel window size has been increased, and both ssh(1)
    sshd(8) now send window updates more aggressively. These improves
    performance on high-BDP (Bandwidth Delay Product) networks.

  * ssh(1) and sshd(8) now preserve MAC contexts between packets, which
    saves 2 hash calls per packet and results in 12-16% speedup for
    arcfour256/hmac-md5.

  * A new MAC algorithm has been added, UMAC-64 (RFC4418) as
    "umac-64@openssh.com". UMAC-64 has been measured to be
    approximately 20% faster than HMAC-MD5.

  * A -K flag was added to ssh(1) to set GSSAPIAuthentication=Yes

  * Failure to establish a ssh(1) TunnelForward is now treated as a
    fatal error when the ExitOnForwardFailure option is set.

  * ssh(1) returns a sensible exit status if the control master goes
    away without passing the full exit status. (bz #1261)

  * The following bugs have been fixed in this release:

    - When using a ProxyCommand in ssh(1), set the outgoing hostname with
      gethostname(2), allowing hostbased authentication to work (bz #616)
    - Make scp(1) skip FIFOs rather than hanging (bz #856)
    - Encode non-printing characters in scp(1) filenames.
      these could cause copies to be aborted with a "protocol error"
      (bz #891)
    - Handle SIGINT in sshd(8) privilege separation child process to
      ensure that wtmp and lastlog records are correctly updated
      (bz #1196)
    - Report GSSAPI mechanism in errors, for libraries that support
      multiple mechanisms (bz #1220)
    - Improve documentation for ssh-add(1)'s -d option (bz #1224)
    - Rearrange and tidy GSSAPI code, removing server-only code being
      linked into the client. (bz #1225)
    - Delay execution of ssh(1)'s LocalCommand until after all forwadings
      have been established. (bz #1232)
    - In scp(1), do not truncate non-regular files (bz #1236)
    - Improve exit message from ControlMaster clients. (bz #1262)
    - Prevent sftp-server(8) from reading until it runs out of buffer
      space, whereupon it would exit with a fatal error. (bz #1286)

  * Portable OpenSSH bugs fixed:

    - Fix multiple inclusion of paths.h on AIX 5.1 systems. (bz #1243)
    - Implement getpeereid for Solaris using getpeerucred. Solaris
      systems will now refuse ssh-agent(1) and ssh(1) ControlMaster
      clients from different, non-root users (bz #1287)
    - Fix compilation warnings by including string.h if found. (bz #1294)
    - Remove redefinition of _res in getrrsetbyname.c for platforms that
      already define it. (bz #1299)
    - Fix spurious "chan_read_failed for istate 3" errors from sshd(8),
      a side-effect of the "hang on exit" fix introduced in 4.6p1.
      (bz #1306)
    - pam_end() was not being called if authentication failed (bz #1322)
    - Fix SELinux support when SELinux is in permissive mode. Previously
      sshd(8) was treating SELinux errors as always fatal. (bz #1325)
    - Ensure that pam_setcred(..., PAM_ESTABLISH_CRED) is called before
      pam_setcred(..., PAM_REINITIALIZE_CRED), fixing pam_dhkeys.
      (bz #1339)
    - Fix privilege separation on QNX - pre-auth only, this platform does
      not support file descriptior passing needed for post-auth privilege
      separation. (bz #1343)

Thanks to everyone who has contributed patches, reported bugs and
tested releases.

Checksums:
==========

- SHA1 (openssh-4.7.tar.gz) = 9ebaab9b31e01bd0d04425dc23536bcc78f8d990
- SHA1 (openssh-4.7p1.tar.gz) = 58357db9e64ba6382bef3d73d1d386fcdc0508f4

Reporting Bugs:
===============

- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 4.6/4.6p1 (2007-03-09)

```
OpenSSH 4.6 was released on 2007-03-09. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots and purchased
T-shirts or posters.
```

```
T-shirt, poster and CD sales directly support the project. Pictures
and more information can be found at:
        http://www.openbsd.org/tshirts.html and
        http://www.openbsd.org/orders.html

For international orders use http://https.openbsd.org/cgi-bin/order
and for European orders, use http://https.openbsd.org/cgi-bin/order.eu

Changes since OpenSSH 4.5:
============================

 * sshd now allows the enabling and disabling of authentication
   methods on a per user, group, host and network basis via the
   Match directive in sshd_config.

 * The following bugs have been fixed in this release:

   - Clear SIGALRM when restarting due to SIGHUP. Prevents stray
     signal from taking down sshd if a connection was pending at
     the time SIGHUP was received
   - sftp returned a zero exit status when upload failed due to write
     errors (bugzilla #1252)
   - fixed an inconsistent check for a terminal when displaying scp
     progress meter (bugzilla #1265)
   - Parsing of time values in Match blocks was incorrectly applied
     to the global configuration (bugzilla #1275)
   - Allow multiple forwarding options to work when specified in a
     PermitOpen directive (bugzilla #1267)
   - Interoperate with ssh.com versions that do not support binding
     remote port forwarding sessions to a hostname (bugzilla #1019)

 * Portable OpenSSH bugs fixed:

   - "hang on exit" when background processes are running at the time
     of exit on a ttyful/login session (bugzilla #52)
   - Fix typos in the ssh-rand-helper(8) man page (bugzilla #1259)
   - Check that some SIG records have been returned in getrrsetbyname
     (bugzilla #1281)
   - Fix contrib/findssl for platforms that lack "which" (bugzilla
     #1237)
   - Work around bug in OpenSSL 0.9.8e that broke aes256-ctr,
     aes192-ctr, arcfour256 (bugzilla #1291)

Checksums:
==========

- SHA1 (openssh-4.6.tar.gz) = c1700845be464a769428f34ef727c1f530728afc
- SHA1 (openssh-4.6p1.tar.gz) = b2aefeb1861b4688b1777436035239ec32a47da8


Reporting Bugs:
===============

- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 4.5/4.5p1 (2006-11-07)

```
OpenSSH 4.5 was released on 2006-11-07. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots and purchased
T-shirts or posters.

T-shirt, poster and CD sales directly support the project. Pictures
and more information can be found at:
        http://www.openbsd.org/tshirts.html and
        http://www.openbsd.org/orders.html

For international orders use http://https.openbsd.org/cgi-bin/order
and for European orders, use http://https.openbsd.org/cgi-bin/order.eu

Changes since OpenSSH 4.4:
============================

This is a bugfix only release. No new features have been added.

Security bugs resolved in this release:

 * Fix a bug in the sshd privilege separation monitor that weakened its
   verification of successful authentication. This bug is not known to
   be exploitable in the absence of additional vulnerabilities.

This release includes the following non-security fixes:

 * Several compilation fixes for portable OpenSSH
```

```
 * Fixes to Solaris SMF/process contract support (bugzilla #1255)

Thanks to everyone who has contributed patches, reported bugs and
tested releases.

Checksums:
==========

- SHA1 (openssh-4.5.tar.gz) = def3de1557181062d788695b9371d02635af39fb
- SHA1 (openssh-4.5p1.tar.gz) = 2eefcbbeb9e4fa16fa4500dec107d1a09d3d02d7

Reporting Bugs:
===============

- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 4.4/4.4p1 (2006-09-27)

```
OpenSSH 4.4 was released on 2006-09-27. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots and purchased
T-shirts or posters.

T-shirt, poster and CD sales directly support the project. Pictures
and more information can be found at:
        http://www.openbsd.org/tshirts.html and
        http://www.openbsd.org/orders.html

For international orders use http://https.openbsd.org/cgi-bin/order
and for European orders, use http://https.openbsd.org/cgi-bin/order.eu

Changes since OpenSSH 4.3:
==========================

Security bugs resolved in this release:

 * Fix a pre-authentication denial of service found by Tavis Ormandy,
   that would cause sshd(8) to spin until the login grace time
   expired.

 * Fix an unsafe signal hander reported by Mark Dowd. The signal
   handler was vulnerable to a race condition that could be exploited
   to perform a pre-authentication denial of service. On portable
   OpenSSH, this vulnerability could theoretically lead to
   pre-authentication remote code execution if GSSAPI authentication
   is enabled, but the likelihood of successful exploitation appears
   remote.

 * On portable OpenSSH, fix a GSSAPI authentication abort that could
   be used to determine the validity of usernames on some platforms.

This release includes the following new functionality and fixes:

 * Implemented conditional configuration in sshd_config(5) using the
   "Match" directive. This allows some configuration options to be
   selectively overridden if specific criteria (based on user, group,
   hostname and/or address) are met. So far a useful subset of post-
   authentication options are supported and more are expected to be
   added in future releases.

 * Add support for Diffie-Hellman group exchange key agreement with a
   final hash of SHA256.

 * Added a "ForceCommand" directive to sshd_config(5). Similar to the
   command="..." option accepted in ~/.ssh/authorized_keys, this forces
   the execution of the specified command regardless of what the user
   requested. This is very useful in conjunction with the new "Match"
   option.

 * Add a "PermitOpen" directive to sshd_config(5). This mirrors the
   permitopen="..." authorized_keys option, allowing fine-grained
   control over the port-forwardings that a user is allowed to
   establish.

 * Add optional logging of transactions to sftp-server(8).

 * ssh(1) will now record port numbers for hosts stored in
   ~/.ssh/authorized_keys when a non-standard port has been requested.

 * Add an "ExitOnForwardFailure" option to cause ssh(1) to exit (with
   a non-zero exit code) when requested port forwardings could not be
   established.

 * Extend sshd_config(5) "SubSystem" declarations to allow the
```

```
    specification of command-line arguments.

  * Replacement of all integer overflow susceptible invocations of
    malloc(3) and realloc(3) with overflow-checking equivalents.

  * Many manpage fixes and improvements

  * New portable OpenSSH-specific features:

    - Add optional support for SELinux, controlled using the
      --with-selinux configure option (experimental)

    - Add optional support for Solaris process contracts, enabled
      using the --with-solaris-contracts configure option (experimental)
      This option will also include SMF metadata in Solaris packages
      built using the "make package" target

    - Add optional support for OpenSSL hardware accelerators (engines),
      enabled using the --with-ssl-engine configure option.

  * Bugs from http://bugzilla.mindrot.org fixed:
     #482  - readconf doesn't accept paths with spaces in them.
     #906  - syslog messages from sshd [net] lost.
     #975  - Kerberos authentication timing can leak information
             about account validity.
     #981  - Flow stop in SSH2.
     #1102 - C program 'write' with zero length hangs.
     #1129 - sshd hangs for command-only invocations due to
             fork/child signals.
     #1131 - error "buffer_append_space:alloc not supported"
     #1138 - Passphrase asked for (but ignored) if key file permissions
             too liberal..
     #1156 - Closes connection after C-c is pressed on QNX.
     #1157 - ssh-keygen doesn't handle DOS line breaks.
     #1159 - %u and %h not handled in IdentityFile.
     #1161 - scp -r fails.
     #1162 - Inappropriate sequence of syslog messages.
     #1166 - openssh-4.3p1 has some issues compiling.
     #1171 - configure can't always figure out LLONG_MAX..
     #1173 - scp reports lost connection for very large files.
     #1177 - Incorrect sshrc file location in Makefile.in.
     #1179 - sshd incorrectly rejects  connections due to IP options.
     #1181 - configure should detect when openssl-0.9.8x needs -ldl.
     #1186 - ssh tries multiple times to open unprotected keys.
     #1188 - keyboard-interactive should not allow retry after
             pam_acct_mgmt fails.
     #1193 - Open ssh will not allow changing of passwords on usernames
             greater than 8 characters..
     #1201 - Bind address information is not specified in command line
             help messages.
     #1203 - configure.ac is missing an open [.
     #1207 - sshd does not clear unsuccessful login count on
             non-interactive logins.
     #1218 - GSSAPI client code permits SPNEGO usage.
     #1221 - Banner only suppressed at log level = QUIET (used to be
             at log level < INFO).

  * Fixes to memory and file descriptor leaks reported by the Coverity
    static analysis tool

  * Fixes to inconsistent pointer checks reported by the Stanford
    SATURN tool

Thanks to everyone who has contributed patches, reported bugs and
tested releases.

Checksums:
==========

- SHA1 (openssh-4.4.tar.gz) = 2294b5e5a591420aa05ff607c1890ab622ace878
- SHA1 (openssh-4.4p1.tar.gz) = 6a52b1dee1c2c9862923c0008d201d98a7fd9d6c

Reporting Bugs:
===============

- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 4.3p2 (2006-02-11)

```
Portable OpenSSH 4.3p2 was released on 2006-02-11. It will be available
from the mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

We have also recently completed another Internet SSH usage scan, the
results of which may be found at http://www.openssh.com/usage.html

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots and purchased
```

T-shirts or posters.

T-shirt, poster and CD sales directly support the project. Pictures
and more information can be found at:
        http://www.openbsd.org/tshirts.html and
        http://www.openbsd.org/orders.html

For international orders use http://https.openbsd.org/cgi-bin/order
and for European orders, use http://https.openbsd.org/cgi-bin/order.eu

Changes since Portable OpenSSH 4.3p1:
=====================================

This is a release of Portable OpenSSH only, to resolve some
portability bugs. There are no new features, only fixes:

 * Explicitly test for egrep in ./configure, fixing a problem in 4.3p1
   that caused some platforms to fail to detect the available fields
   in utmp/wtmp/lastlog records. This bug manifested as missing or
   empty login/logout records (as seen by last(1), etc.)

 * Fix for logout records not being updated on platforms without
   support for post-authentication privilege separation (e.g. Cygwin)

 * Fixed compilation problems on Ultrix, NewsOS and QNX

Thanks to everyone who has contributed patches, reported bugs or test
releases.

Checksums:
==========

- SHA1 (openssh-4.3p2.tar.gz) = 2b5b0751fd578283ba7b106025c0ba391fd72f1f

Reporting Bugs:
===============

- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.

OpenSSH 4.3/4.3p1 (2006-02-01)

OpenSSH 4.3 was released on 2006-02-01. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

We have also recently completed another Internet SSH usage scan, the
results of which may be found at http://www.openssh.com/usage.html

Once again, we would like to thank the OpenSSH community for their
continued support of the project, especially those who contributed
code or patches, reported bugs, tested snapshots and purchased
T-shirts or posters.

T-shirt, poster and CD sales directly support the project. Pictures
and more information can be found at:
        http://www.openbsd.org/tshirts.html and
        http://www.openbsd.org/orders.html

For international orders use http://https.openbsd.org/cgi-bin/order
and for European orders, use http://https.openbsd.org/cgi-bin/order.eu

Changes since OpenSSH 4.2:
==========================

Security bugs resolved in this release:

 * CVE-2006-0225: scp (as does rcp, on which it is based) invoked a
   subshell to perform local to local, and remote to remote copy
   operations. This subshell exposed filenames to shell expansion
   twice; allowing a local attacker to create filenames containing
   shell metacharacters that, if matched by a wildcard, could lead
   to execution of attacker-specified commands with the privilege of
   the user running scp (Bugzilla #1094)

This is primarily a bug-fix release, only one new feature has been
added:

 * Add support for tunneling arbitrary network packets over a
   connection between an OpenSSH client and server via tun(4) virtual
   network interfaces. This allows the use of OpenSSH (4.3+) to create
   a true VPN between the client and server providing real network
   connectivity at layer 2 or 3. This feature is experimental and is
   currently supported on OpenBSD, Linux, NetBSD (IPv4 only) and
   FreeBSD. Other operating systems with tun/tap interface capability
   may be added in future portable OpenSSH releases. Please refer to
   the README.tun file in the source distribution for further details
   and usage examples.

Some of the other bugs resolved and internal improvements are:

```
  * Reduce default key length for new DSA keys generated by ssh-keygen
    back to 1024 bits. DSA is not specified for longer lengths and does
    not fully benefit from simply making keys longer. As per FIPS 186-2
    Change Notice 1, ssh-keygen will refuse to generate a new DSA key
    smaller or larger than 1024 bits

  * Fixed X forwarding failing to start when a the X11 client is executed
    in background at the time of session exit (Bugzilla #1086)

  * Change ssh-keygen to generate a protocol 2 RSA key when invoked
    without arguments (Bugzilla #1064)

  * Fix timing variance for valid vs. invalid accounts when attempting
    Kerberos authentication (Bugzilla #975)

  * Ensure that ssh always returns code 255 on internal error (Bugzilla
    #1137)

  * Cleanup wtmp files on SIGTERM when not using privsep (Bugzilla #1029)

  * Set SO_REUSEADDR on X11 listeners to avoid problems caused by
    lingering sockets from previous session (X11 applications can
    sometimes not connect to 127.0.0.1:60xx) (Bugzilla #1076)

  * Ensure that fds 0, 1 and 2 are always attached in all programs, by
    duping /dev/null to them if necessary.

  * Xauth list invocation had bogus "." argument (Bugzilla #1082)

  * Remove internal assumptions on key exchange hash algorithm and output
    length, preparing OpenSSH for KEX methods with alternate hashes.

  * Ignore junk sent by a server before  it sends the "SSH-" banner
    (Bugzilla #1067)

  * The manpages has been significantly improves and rearranged, in
    addition to other specific manpage fixes:
    #1037 - Man page entries for -L and -R should mention -g.
    #1077 - Descriptions for "ssh -D" and DynamicForward should mention
            they can specify "bind_address" optionally.
    #1088 - Incorrect descriptions in ssh_config man page for
            ControlMaster=no.
    #1121 - Several corrections for ssh_agent manpages

  * Lots of cleanups, including fixes to memory leaks on error paths
    (Bugzilla #1109, #1110, #1111 and more) and possible crashes (#1092)

  * Portable OpenSSH-specific fixes:

    - Pass random seed during re-exec for each connection: speeds up
      processing of new connections on platforms using the OpenSSH's
      builtin entropy collector (ssh-rand-helper)

    - PAM fixes and improvements:
      #1045 - Missing option for ignoring the /etc/nologin file
      #1087 - Show PAM password expiry message from LDAP on login
      #1028 - Forward final non-query conversations to client
      #1126 - Prevent user from being forced to change an expired
              password repeatedly on AIX in some PAM configurations.
      #1045 - Do not check /etc/nologin when PAM is enabled, instead
              allow PAM to handle it.  Note that on platforms using
              PAM, the pam_nologin module should be used in sshd's
              session stack in order to maintain past behaviour

    - Portability-related fixes:
      #989  - Fix multiplexing regress test on Solaris
      #1097 - Cross-compile fixes.
      #1096 - ssh-keygen broken on HPUX.
      #1098 - $MAIL being set incorrectly for HPUX server login.
      #1104 - Compile error on Tru64 Unix 4.0f
      #1106 - Updated .spec file and startup for SuSE.
      #1122 - Use _GNU_SOURCE define in favor of __USE_GNU, fixing
              compilation problems on glibc 2.4

Thanks to everyone who has contributed patches, reported bugs or test
releases.

Checksums:
==========

- SHA1 (openssh-4.3.tar.gz) = 0cb66e56805d66b51511455423bab88aa58a1455
- SHA1 (openssh-4.3p1.tar.gz) = b1f379127829e7e820955b2825130edd1601ba59

Reporting Bugs:
===============

- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Darren Tucker, Jason McIntyre, Tim Rice and
Ben Lindstrom.
```

OpenSSH 4.2/4.2p1 (2005-09-01)

OpenSSH 4.2 was released on 2005-09-01. It is available from the mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0 implementation and includes sftp client and server support.

We would like to thank the OpenSSH community for their continued support of the project, especially those who contributed source, reported bugs, tested snapshots and purchased T-shirts or posters.

T-shirt, poster and CD sales directly support the project. Pictures and more information can be found at:
http://www.openbsd.org/tshirts.html and
http://www.openbsd.org/orders.html

For international orders use http://https.openbsd.org/cgi-bin/order and for European orders, use http://https.openbsd.org/cgi-bin/order.eu

Changes since OpenSSH 4.1:
==========

- SECURITY: Fix a bug introduced in OpenSSH 4.0 that caused GatewayPorts to be incorrectly activated for dynamic ("-D") port forwardings when no listen address was explicitly specified.

- SECURITY: sshd in OpenSSH versions prior to 4.2 allow GSSAPI credentials to be delegated to users who log in with methods other than GSSAPI authentication (e.g. public key) when the client requests it. This behaviour has been changed in OpenSSH 4.2 to only delegate credentials to users who authenticate using the GSSAPI method. This eliminates the risk of credentials being inadvertently exposed to an untrusted user/host (though users should not activate GSSAPIDelegateCredentials to begin with when the remote user or host is untrusted)

- Added a new compression method that delays the start of zlib compression until the user has been authenticated successfully. The new method ("Compression delayed") is on by default in the server. This eliminates the risk of any zlib vulnerability leading to a compromise of the server from unauthenticated users.

  NB. Older OpenSSH (<3.5) versions have a bug that will cause them to refuse to connect to any server that does not offer compression when the client has compression requested. Since the new "delayed" server mode isn't supported by these older clients, they will refuse to connect to a new server unless compression is disabled (on the client end) or the original compression method is enabled on the server ("Compression yes" in sshd_config)

- Another round of proactive changes for signed vs unsigned integer bugs has been completed, including changing the atomicio() API to encourage safer programming. This work is ongoing.

- Added support for the improved arcfour cipher modes from draft-harris-ssh-arcfour-fixes-02. The improves the cipher's resistance to a number of attacks by discarding early keystream output.

- Increase the default size of new RSA/DSA keys generated by ssh-keygen from 1024 to 2048 bits.

- Many bugfixes and improvements to connection multiplexing, including:

  - Added ControlMaster=auto/autoask options to support opportunistic multiplexing (see the ssh_config(5) manpage for details).

  - The client will now gracefully fallback to starting a new TCP connection if it cannot connect to a specified multiplexing control socket

  - Added %h (target hostname), %p (target port) and %r (remote username) expansion sequences to ControlPath. Also allow ControlPath=none to disable connection multiplexing.

  - Implemented support for X11 and agent forwarding over multiplexed connections. Because of protocol limitations, the slave connections inherit the master's DISPLAY and SSH_AUTH_SOCK rather than distinctly forwarding their own.

- Portable OpenSSH: Added support for long passwords (> 8-char) on UnixWare 7.

- The following bugs from http://bugzilla.mindrot.org/ were closed:

  #471 - Misleading error message if /dev/tty perms wrong
  #623 - Don't use $HOME in manpages
  #829 - Don't allocate a tty if -n option is set
  #1025 - Correctly handle disabled special character in ttymodes
  #1033 - Fix compile-time warnings
  #1046 - AIX 5.3 Garbage on Login
  #1054 - Don't terminate connection on getpeername() failure
  #1076 - GSSAPIDelegateCredentials issue mentioned above

- Lots of other improvements and fixes. Please refer to the ChangeLog for details

Thanks to everyone who has contributed patches, problem or test reports.

```
Checksums:
==========

- SHA1 (openssh-4.2.tar.gz) = d2bd777986a30e446268ceeb24cddbf2edf51b21
- SHA1 (openssh-4.2p1.tar.gz) = 5e7231cfa8ec673ea856ce291b78fac8b380eb78

Reporting Bugs:
===============

- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Ben Lindstrom, Darren Tucker and Tim Rice.
```

OpenSSH 4.1/4.1p1 (2005-05-26)

```
OpenSSH 4.1 was released on 2005-05-26. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

We would like to thank the OpenSSH community for their continued
support to the project, especially those who contributed source and
bought T-shirts or posters.

We have a new design of T-shirt available, more info on
        http://www.openbsd.org/tshirts.html#18

For international orders use http://https.openbsd.org/cgi-bin/order
and for European orders, use http://https.openbsd.org/cgi-bin/order.eu


Changes since OpenSSH 4.0:
==========================

* This is a bugfix release, no new features have been added. Some notable
  fixes are:

  - Fix segfault when using forwardings configured in ssh_config(5) and
    ClearAllForwardings (bugzilla #996)

  - Limit input buffer size for channels. A peer could send more data
    than the buffer code was willing to accept. This would cause OpenSSH
    to abort the connection (bugzilla #896)

* Several improvements to the regression tests

* Portable OpenSSH:

  - OpenSSH will now always normalise IPv4 in IPv6 mapped addresses back to
    IPv4 addresses. This means that IPv4 addresses in log messages on IPv6
    enabled machines will no longer be prefixed by "::ffff:" and AllowUsers,
    DenyUsers, AllowGroups, DenyGroups will match IPv4-style addresses only
    for 4-in-6 mapped connections. This ensures a consistent representation
    of IPv4 addresses regardless of whether or not the machine is IPv6
    enabled.

* Other bugfixes, including bugzilla #950, #997, #998, #999, #1005, #1006,
  #1024, and #1038

Changes since OpenSSH 3.9:
==========================

* ssh(1) now allows the optional specification of an address to bind to
  in port forwarding connections (local, remote and dynamic). Please
  refer to the documentation for the -L and -R options in the ssh(1)
  manual page and the LocalForward and RemoteForward options in the
  ssh_config(5) manpage. (Bugzilla #413)

* To control remote bindings while retaining backwards compatibility,
  sshd(8)'s GatewayPorts option has been extended. To allow client
  specified bind addresses for remote (-R) port forwardings, the server
  must be configured with "GatewayPorts clientspecified".

* ssh(1) and ssh-keyscan(1) now support hashing of host names and
  addresses added to known_hosts files, controlled by the ssh(1)
  HashKnownHosts configuration directive. This option improves user
  privacy by hiding which hosts have been visited. At present this
  option is off by default, but may be turned on once it receives
  sufficient testing.

* Added options for managing keys in known_hosts files to ssh-keygen(1),
  including the ability to search for hosts by name, delete hosts by
  name and convert an unhashed known_hosts file into one with hashed
  names. These are particularly useful for managing known_hosts files
  with hashed hostnames.

* Improve account and password expiry support in sshd(8). Ther server
  will now warn in advance for both account and password expiry.

* sshd(8) will now log the source of connections denied by AllowUsers,
  DenyUsers, AllowGroups and DenyGroups (Bugzilla #909)

* Added AddressFamily option to sshd(8) to allow global control over
```

IPv4/IPv6 usage. (Bugzilla #989)

* Improved sftp(1) client, including bugfixes and optimisations for the ``ls'' command and command history and editing support using libedit.

* Improved the handling of bad data in authorized_keys files, eliminating fatal errors on corrupt or very large keys. (Bugzilla #884)

* Improved connection multiplexing support in ssh(1). Several bugs have been fixed and a new "command mode" has been added to allow the control of a running multiplexing master connection, including checking that it is up, determining its PID and asking it to exit.

* Have scp(1) and sftp(1) wait for the spawned ssh to exit before they exit themselves. This prevents ssh from being unable to restore terminal modes (not normally a problem on OpenBSD but common with -Portable on POSIX platforms). (Bugzilla #950)

* Portable OpenSSH:

  - Add *EXPERIMENTAL* BSM audit support for Solaris systems (Bugzilla #125)

  - Enable IPv6 on AIX where possible (see README.platform for details), working around a misfeature of AIX's getnameinfo. (Bugzilla #835)

  - Teach sshd(8) to write failed login records to btmp for unsuccessful auth attempts. Currently this is only for password, keyboard-interactive and challenge/response authentication methods and only on Linux and HP-UX.

  - sshd(8) now sends output from failing PAM session modules to the user before exiting, similar to the way /etc/nologin is handled

  - Store credentials from gssapi-with-mic authentication early enough to be available to PAM session modules when privsep=yes.

Checksums:
==========

- SHA1 (openssh-4.1.tar.gz) = 62fc9596b20244bb559d5fee3ff3ecc0dfd557cb
- SHA1 (openssh-4.1p1.tar.gz) = e85d389da8ad8290f5031b8f9972e2623c674e46

Reporting Bugs:
===============

- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt, Kevin Steves, Damien Miller, Ben Lindstrom, Darren Tucker and Tim Rice.

OpenSSH 4.0/4.0p1 (2005-03-09)

OpenSSH 4.0 was released on 2005-03-09. It is available from the mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0 implementation and includes sftp client and server support.

We would like to thank the OpenSSH community for their continued support to the project, especially those who contributed source and bought T-shirts or posters.

We have a new design of T-shirt available, more info on
        http://www.openbsd.org/tshirts.html#18

For international orders use http://https.openbsd.org/cgi-bin/order
and for European orders, use http://https.openbsd.org/cgi-bin/order.eu

Changes since OpenSSH 3.9:
==========================

* ssh(1) now allows the optional specification of an address to bind to in port forwarding connections (local, remote and dynamic). Please refer to the documentation for the -L and -R options in the ssh(1) manual page and the LocalForward and RemoteForward options in the ssh_config(5) manpage. (Bugzilla #413)

* To control remote bindings while retaining backwards compatibility, sshd(8)'s GatewayPorts option has been extended. To allow client specified bind addresses for remote (-R) port forwardings, the server must be configured with "GatewayPorts clientspecified".

* ssh(1) and ssh-keyscan(1) now support hashing of host names and addresses added to known_hosts files, controlled by the ssh(1) HashKnownHosts configuration directive. This option improves user privacy by hiding which hosts have been visited. At present this option is off by default, but may be turned on once it receives sufficient testing.

* Added options for managing keys in known_hosts files to ssh-keygen(1), including the ability to search for hosts by name, delete hosts by name and convert an unhashed known_hosts file into one with hashed

```
    names. These are particularly useful for managing known_hosts files
    with hashed hostnames.

  * Improve account and password expiry support in sshd(8). Ther server
    will now warn in advance for both account and password expiry.

  * sshd(8) will now log the source of connections denied by AllowUsers,
    DenyUsers, AllowGroups and DenyGroups (Bugzilla #909)

  * Added AddressFamily option to sshd(8) to allow global control over
    IPv4/IPv6 usage. (Bugzilla #989)

  * Improved sftp(1) client, including bugfixes and optimisations for the
    ``ls'' command and command history and editing support using libedit.

  * Improved the handling of bad data in authorized_keys files,
    eliminating fatal errors on corrupt or very large keys. (Bugzilla
    #884)

  * Improved connection multiplexing support in ssh(1). Several bugs
    have been fixed and a new "command mode" has been added to allow the
    control of a running multiplexing master connection, including
    checking that it is up, determining its PID and asking it to exit.

  * Have scp(1) and sftp(1) wait for the spawned ssh to exit before they
    exit themselves.  This prevents ssh from being unable to restore
    terminal modes (not normally a problem on OpenBSD but common with
    -Portable on POSIX platforms). (Bugzilla #950)

  * Portable OpenSSH:

    - Add *EXPERIMENTAL* BSM audit support for Solaris systems
      (Bugzilla #125)

    - Enable IPv6 on AIX where possible (see README.platform for
      details), working around a misfeature of AIX's getnameinfo.
      (Bugzilla #835)

    - Teach sshd(8) to write failed login records to btmp for
      unsuccessful auth attempts. Currently this is only for password,
      keyboard-interactive and challenge/response authentication methods
      and only on Linux and HP-UX.

    - sshd(8) now sends output from failing PAM session modules to the
      user before exiting, similar to the way /etc/nologin is handled

    - Store credentials from gssapi-with-mic authentication early enough
      to be available to PAM session modules when privsep=yes.

Checksums:
==========

- MD5 (openssh-4.0.tgz) = 7dbf15fe7c294672e8822127f50107d0
- MD5 (openssh-4.0p1.tar.gz) = 7b36f28fc16e1b7f4ba3c1dca191ac92

Reporting Bugs:
===============

- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Ben Lindstrom, Darren Tucker and Tim Rice.
```

OpenSSH 3.9/3.9p1 (2004-08-18)

```
OpenSSH 3.9 was released on 2004-08-18. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

We would like to thank the OpenSSH community for their continued
support to the project, especially those who contributed source and
bought T-shirts or posters.

We have a new design of T-shirt available, more info on
        http://www.openbsd.org/tshirts.html#18

For international orders use http://https.openbsd.org/cgi-bin/order
and for European orders, use http://https.openbsd.org/cgi-bin/order.eu


Changes since OpenSSH 3.8:
==========================

  * Added new "IdentitiesOnly" option to ssh(1), which specifies that it should
    use keys specified in ssh_config, rather than any keys in ssh-agent(1)

  * Make sshd(8) re-execute itself on accepting a new connection. This security
    measure ensures that all execute-time randomisations are reapplied for each
    connection rather than once, for the master process' lifetime. This includes
    mmap and malloc mappings, shared library addressing, shared library mapping
    order, ProPolice and StackGhost cookies on systems that support such things

  * Add strict permission and ownership checks to programs reading ~/.ssh/config
    NB ssh(1) will now exit instead of trying to process a config with poor
```

ownership or permissions

* Implemented the ability to pass selected environment variables between the client and the server. See "AcceptEnv" in sshd_config(5) and "SendEnv" in ssh_config(5) for details

* Added a "MaxAuthTries" option to sshd(8), allowing control over the maximum number of authentication attempts permitted per connection

* Added support for cancellation of active remote port forwarding sessions. This may be performed using the ~C escape character, see "Escape Characters" in ssh(1) for details

* Many sftp(1) interface improvements, including greatly enhanced "ls" support and the ability to cancel active transfers using SIGINT (^C)

* Implement session multiplexing: a single ssh(1) connection can now carry multiple login/command/file transfer sessions. Refer to the "ControlMaster" and "ControlPath" options in ssh_config(5) for more information

* The sftp-server has improved support for non-POSIX filesystems (e.g. FAT)

* Portable OpenSSH: Re-introduce support for PAM password authentication, in addition to the keyboard-interactive driver. PAM password authentication is less flexible, and doesn't support pre-authentication password expiry but runs in-process so Kerberos tokens, etc are retained

* Improved and more extensive regression tests

* Many bugfixes and small improvements

Checksums:
==========

- MD5 (openssh-3.9.tgz) = 93f48bfcc1560895ae53de6bfc41689b
- MD5 (openssh-3.9p1.tar.gz) = 8e1774d0b52aff08f817f3987442a16e

Reporting Bugs:
===============

- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Ben Lindstrom, Darren Tucker and Tim Rice.

OpenSSH 3.8.1p1 (2004-04-19)

OpenSSH 3.8.1p1 was released on 2004-04-19. It is available from
the mirrors listed at https://www.openssh.com/.

This release is a bug-fix release for the portable version. There are
no feature additions and no corresponding OpenBSD-only release.

OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

We would like to thank the OpenSSH community for their continued
support to the project, especially those who contributed source, help
with testing and have bought T-shirts or posters.

We have a new design of T-shirt available, more info on
        http://www.openbsd.org/tshirts.html#18

For international orders use http://https.openbsd.org/cgi-bin/order
and for European orders, use http://https.openbsd.org/cgi-bin/order.eu

Bugs fixed since OpenSSH 3.8p1:
===============================

Bug #673 - Fix compilation on NetBSD with S/Key enabled

Bug #748 - Detect and workaround broken name resolution on HP-UX

Bug #802 - Fix linking on Tru64 when compiled with SIA support

Bug #808 - Fix PAM crash on expired password when not authenticated using
           pam/kbdint mechanism

Bug #810 - Fix erroneous clearing of TZ environment variable

Bug #811 - Improve locked password detection across Linux variants

Bug #820 - Fix utmp corruption on Irix

Bug #825 - Fix disconnection problem when using IPv4-in-IPv6 mapped
           addresses on Solaris.

- Fix compilation on OS X systems with Kerberos/GSSAPI

- Many more minor fixes, please refer to the ChangeLog file for details

Checksums:
==========

- MD5 (openssh-3.8.1p1.tar.gz) = 1dbfd40ae683f822ae917eebf171ca42

Reporting Bugs:
===============

- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Ben Lindstrom, Darren Tucker and Tim Rice.

OpenSSH 3.8.1/3.8.1p1

3.8.1p1 is a Portable-only release, there is no corresponding 3.8.1 release.
See http://www.openssh.com/txt/release-3.8.1p1 for the Portable release notes.

OpenSSH 3.8/3.8p1 (2004-02-24)

OpenSSH 3.8 was released on 2004-02-24. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

We would like to thank the OpenSSH community for their continued
support to the project, especially those who contributed source and
bought T-shirts or posters.

We have a new design of T-shirt available, more info on
        http://www.openbsd.org/tshirts.html#18

For international orders use http://https.openbsd.org/cgi-bin/order
and for European orders, use http://https.openbsd.org/cgi-bin/order.eu

Changes since OpenSSH 3.7.1:
============================

* sshd(8) now supports forced changes of expired passwords via
  /usr/bin/passwd or keyboard-interactive authentication.

  Note for AIX: sshd will now deny password access to accounts with
  passwords expired longer than their maxexpired attribute. For
  details, see the AIX section in README.platform.

* ssh(1) now uses untrusted cookies for X11-Forwarding.
  Some X11 applications might need full access to the X11 server,
  see ForwardX11Trusted in ssh(1) and xauth(1) for more information.

* ssh(1) now supports sending application layer
  keep-alive messages to the server. See ServerAliveInterval
  in ssh(1) for more information.

* Improved sftp(1) batch file support.

* New KerberosGetAFSToken option for sshd(8).

* Updated /etc/moduli file and improved performance for
  protocol version 2.

* Support for host keys in DNS (draft-ietf-secsh-dns-xx.txt).
  Please see README.dns in the source distribution for details.

* Fix a number of memory leaks.

* The experimental "gssapi" support has been replaced with
  the "gssapi-with-mic" to fix possible MITM attacks.
  The two versions are not compatible.

Checksums:
==========

- MD5 (openssh-3.8.tgz) = 7d5590a333d8f8aa1fa6f19e24938700
- MD5 (openssh-3.8p1.tar.gz) = 7861a4c0841ab69a6eec5c747daff6fb

Reporting Bugs:
===============

- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Ben Lindstrom, Darren Tucker and Tim Rice.

OpenSSH 3.7.1p2 (2003-09-23)

Portable OpenSSH 3.7.1p2 was released on 2003-09-23. It will be available
from the mirrors listed at http://www.openssh.com/portable.html shortly.

Please note that this is a release to address issues in the portable

```
version only. The items mentioned below do not affect the OpenBSD
version.

OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

We would like to thank the OpenSSH community for their continued
support to the project, especially those who contributed source and
bought T-shirts or posters.

We have a new design of T-shirt available, more info on
        http://www.openbsd.org/tshirts.html#18

For international orders use http://https.openbsd.org/cgi-bin/order
and for European orders, use http://https.openbsd.org/cgi-bin/order.eu

Security Changes:
=================

  Portable OpenSSH version 3.7p1 and 3.7.1p1 contain multiple
  vulnerabilities in the new PAM authentication code. At least one of
  these bugs is remotely exploitable (under a non-standard
  configuration, with privsep disabled).

  OpenSSH 3.7.1p2 fixes these bugs. Please note that these bugs do not
  exist in OpenBSD's releases of OpenSSH.

Changes since OpenSSH 3.7.1p1:
==============================

* This release disables PAM by default. To enable it, set "UsePAM yes" in
  sshd_config. Due to complexity, inconsistencies in the specification and
  differences between vendors' PAM implementations we recommend that PAM
  be left disabled in sshd_config unless there is a need for its use.
  Sites using only public key or simple password authentication usually
  have little need to enable PAM support.

* This release now requires zlib 1.1.4 to build correctly. Previous
  versions have security problems.

* Fix compilation for versions of OpenSSL before 0.9.6. Some cipher modes
  are not supported for older OpenSSL versions.

* Fix compilation problems on systems with a missing or lacking inet_ntoa()
  function.

* Workaround problems related to unimplemented or broken setresuid/setreuid
  functions on several platforms.

* Fix compilation on older OpenBSD systems.

* Fix handling of password-less authentication (PermitEmptyPasswords=yes)
  that has not worked since the 3.7p1 release.

Checksums:
==========

- MD5 (openssh-3.7.1p2.tar.gz) = 61cf5b059938718308836d00f6764a94


Reporting Bugs:
===============

- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Ben Lindstrom, Darren Tucker and Tim Rice.
```

OpenSSH 3.7.1/3.7.1p1 (2004-02-25)

```
OpenSSH 3.7.1 was released on 2004-02-25. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

We would like to thank the OpenSSH community for their continued
support to the project, especially those who contributed source and
bought T-shirts or posters.

We have a new design of T-shirt available, more info on
        http://www.openbsd.org/tshirts.html#18

For international orders use https://https.openbsd.org/cgi-bin/order
and for European orders, use https://https.openbsd.org/cgi-bin/order.eu

Security Changes:
=================

  All versions of OpenSSH's sshd prior to 3.7.1 contain buffer
  management errors.  It is uncertain whether these errors are
  potentially exploitable, however, we prefer to see bugs
  fixed proactively.

  OpenSSH 3.7 fixed one of these bugs.
```

OpenSSH 3.7.1 fixes more similar bugs.

Changes since OpenSSH 3.6.1:
==========

* The entire OpenSSH code-base has undergone a license review. As a result, all non-ssh1.x code is under a BSD-style license with no advertising requirement. Please refer to README in the source distribution for the exact license terms.

* Rhosts authentication has been removed in ssh(1) and sshd(8).

* Changes in Kerberos support:

  - KerberosV password support now uses a file cache instead of a memory cache.

  - KerberosIV and AFS support has been removed.

  - KerberosV support has been removed from SSH protocol 1.

  - KerberosV password authentication support remains for SSH protocols 1 and 2.

  - This release contains some GSSAPI user authentication support to replace legacy KerberosV authentication support. At present this code is still considered experimental and SHOULD NOT BE USED.

* Changed order that keys are tried in public key authentication. The ssh(1) client tries the keys in the following order:

  1. ssh-agent(1) keys that are found in the ssh_config(5) file
  2. remaining ssh-agent(1) keys
  3. keys that are only listed in the ssh_config(5) file

  This helps when an ssh-agent(1) has many keys, where the sshd(8) server might close the connection before the correct key is tried.

* SOCKS5 support has been added to the dynamic forwarding mode in ssh(1).

* Removed implementation barriers to operation of SSH over SCTP.

* sftp(1) client can now transfer files with quote characters in their filenames.

* Replaced sshd(8)'s VerifyReverseMapping with UseDNS option. When UseDNS option is on, reverse hostname lookups are always performed.

* Fix a number of memory leaks.

* Support for sending tty BREAK over SSH protocol 2.

* Workaround for other vendor bugs in KEX guess handling.

* Support for generating KEX-GEX groups (/etc/moduli) in ssh-keygen(1).

* Automatic re-keying based on amount of data sent over connection.

* New AddressFamily option on client to select protocol to use (IPv4 or IPv6).

* Experimental support for the "aes128-ctr", "aes192-ctr", and "aes256-ctr" ciphers for SSH protocol 2.

* Experimental support for host keys in DNS (draft-ietf-secsh-dns-xx.txt). Please see README.dns in the source distribution for details.

* Portable OpenSSH:

  - Replace PAM password authentication kludge with a more correct PAM challenge-response module from FreeBSD.

  - PAM support may now be enabled/disabled at runtime using the UsePAM directive.

  - Many improvements to the OpenSC smartcard support.

  - Regression tests now work with portable OpenSSH. Please refer to regress/README.regress in the source distribution.

  - On platforms that support it, portable OpenSSH now honors the UMASK, PATH and SUPATH attributes set in /etc/default/login.

  - Deny access to locked accounts, regardless of authentication method in use.

Checksums:
==========

- MD5 (openssh-3.7.1.tgz) = 3d2f1644d6a3d3267e5e2421f1385129
- MD5 (openssh-3.7.1p1.tar.gz) = f54e574e606c08ef63ebb1ab2f7689dc

Reporting Bugs:
==========

```
- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Ben Lindstrom, Darren Tucker and Tim Rice.
```

OpenSSH 3.7/3.7p1 (2003-09-16)

```
OpenSSH 3.7 was released on 2003-09-16. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

We would like to thank the OpenSSH community for their continued
support to the project, especially those who contributed source and
bought T-shirts or posters.

We have a new design of T-shirt available, more info on
        http://www.openbsd.org/tshirts.html#18

For international orders use http://https.openbsd.org/cgi-bin/order
and for European orders, use http://https.openbsd.org/cgi-bin/order.eu

Security Changes:
=================

  All versions of OpenSSH's sshd prior to 3.7 contain a buffer
  management error.  It is uncertain whether this error is
  potentially exploitable, however, we prefer to see bugs
  fixed proactively.

  OpenSSH 3.7 fixes this bug.

Changes since OpenSSH 3.6.1:
============================

* The entire OpenSSH code-base has undergone a license review. As
  a result, all non-ssh1.x code is under a BSD-style license with no
  advertising requirement. Please refer to README in the source
  distribution for the exact license terms.

* Rhosts authentication has been removed in ssh(1) and sshd(8).

* Changes in Kerberos support:

    - KerberosV password support now uses a file cache instead of
      a memory cache.

    - KerberosIV and AFS support has been removed.

    - KerberosV support has been removed from SSH protocol 1.

    - KerberosV password authentication support remains for SSH
      protocols 1 and 2.

    - This release contains some GSSAPI user authentication support
      to replace legacy KerberosV authentication support. At present
      this code is still considered experimental and SHOULD NOT BE
      USED.

* Changed order that keys are tried in public key authentication.
  The ssh(1) client tries the keys in the following order:

     1. ssh-agent(1) keys that are found in the ssh_config(5) file
     2. remaining ssh-agent(1) keys
     3. keys that are only listed in the ssh_config(5) file

  This helps when an ssh-agent(1) has many keys, where the sshd(8)
  server might close the connection before the correct key is tried.

* SOCKS5 support has been added to the dynamic forwarding mode
  in ssh(1).

* Removed implementation barriers to operation of SSH over SCTP.

* sftp(1) client can now transfer files with quote characters in
  their filenames.

* Replaced sshd(8)'s VerifyReverseMapping with UseDNS option.
  When UseDNS option is on, reverse hostname lookups are always
  performed.

* Fix a number of memory leaks.

* Support for sending tty BREAK over SSH protocol 2.

* Workaround for other vendor bugs in KEX guess handling.

* Support for generating KEX-GEX groups (/etc/moduli) in ssh-keygen(1).

* Automatic re-keying based on amount of data sent over connection.

* New AddressFamily option on client to select protocol to use (IPv4
  or IPv6).
```

```
* Experimental support for the "aes128-ctr", "aes192-ctr", and
  "aes256-ctr" ciphers for SSH protocol 2.

* Experimental support for host keys in DNS (draft-ietf-secsh-dns-xx.txt).
  Please see README.dns in the source distribution for details.

* Portable OpenSSH:

    - Replace PAM password authentication kludge with a more correct
      PAM challenge-response module from FreeBSD.

    - PAM support may now be enabled/disabled at runtime using the
      UsePAM directive.

    - Many improvements to the OpenSC smartcard support.

    - Regression tests now work with portable OpenSSH.
      Please refer to regress/README.regress in the source distribution.

    - On platforms that support it, portable OpenSSH now honors the
      UMASK, PATH and SUPATH attributes set in /etc/default/login.

    - Deny access to locked accounts, regardless of authentication
      method in use.

Checksums:
==========

- MD5 (openssh-3.7.tgz) = 86864ecc276c5f75b06d4872a553fa70
- MD5 (openssh-3.7p1.tar.gz) = 77662801ba2a9cadc0ac10054bc6cb37


Reporting Bugs:
===============

- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller, Ben Lindstrom, Darren Tucker and Tim Rice.
```

OpenSSH 3.6.1p2 (2003-04-30)

```
OpenSSH 3.6.1p2 was released on 2003-04-30. It is available from the
mirrors listed at https://www.openssh.com/.This is a release
of the Portable version only.

OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

We would like to thank the OpenSSH community for their continued
support and encouragement.


Changes since OpenSSH 3.6.1p1:
==============================

* Security: corrected linking problem on AIX/gcc. AIX users are
  advised to upgrade immediately. For details, please refer to
  separate advisory (aixgcc.adv).

* Corrected build problems on Irix

* Corrected build problem when building with AFS support

* Merged some changes from Openwall Linux


Checksums:
==========

- MD5 (openssh-3.6p1.tar.gz) = f3879270bffe479e1bd057aa36258696

Reporting Bugs:
===============

- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller and Ben Lindstrom.
```

OpenSSH 3.6.1/3.6.1p1 (2003-04-01)

```
OpenSSH 3.6.1 was released on 2003-04-01. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

We would like to thank the OpenSSH community for their continued
support to the project, especially those who contributed source and
bought T-shirts or posters.

We have a new design of T-shirt available, more info on
```

http://www.openbsd.org/tshirts.html#18

For international orders use http://https.openbsd.org/cgi-bin/order
and for European orders, use http://https.openbsd.org/cgi-bin/order.eu

Changes since OpenSSH 3.6:
==========

* The 'kex guesses' bugfix from OpenSSH 3.6 triggers a bug
  in a few other SSH v2 implementations and causes connections to
  stall. OpenSSH 3.6.1 disables this bugfix when interoperating
  with these implementations.

Changes between OpenSSH 3.5 and OpenSSH 3.6:
==========

* RSA blinding is now used by ssh(1), sshd(8) and ssh-agent(1).
  in order to avoid potential timing attacks against the RSA keys.
  Older versions of OpenSSH have been using RSA blinding in
  ssh-keysign(1) only.

  Please note that there is no evidence that the SSH protocol is
  vulnerable to the OpenSSL/TLS timing attack described in
        http://crypto.stanford.edu/~dabo/papers/ssl-timing.pdf

* ssh-agent(1) optionally requires user confirmation if a key gets
  used, see '-c' in ssh-add(1).

* sshd(8) now handles PermitRootLogin correctly when UsePrivilegeSeparation
  is enabled.

* sshd(8) now removes X11 cookies when a session gets closed.

* ssh-keysign(8) is disabled by default and only enabled if the
  new EnableSSHKeysign option is set in the global ssh_config(5)
  file.

* ssh(1) and sshd(8) now handle 'kex guesses' correctly (key exchange
  guesses).

* ssh(1) no longer overwrites SIG_IGN. This matches behaviour from
  rsh(1) and is used by backup tools.

* setting ProxyCommand to 'none' disables the proxy feature, see
  ssh_config(5).

* scp(1) supports add -1 and -2.

* scp(1) supports bandwidth limiting.

* sftp(1) displays a progressmeter.

* sftp(1) has improved error handling for scripting.

Checksums:
==========

- MD5 (openssh-3.6.1p1.tar.gz) = d4c2c88b883f097fe88e327cbb4b2e2a
- MD5 (openssh-3.6.1.tgz) = aa2acd2be17dc3fd514a1e09336aab51

Reporting Bugs:
==========

- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller and Ben Lindstrom.

OpenSSH 3.6/3.6p1 (2003-03-31)

OpenSSH 3.6 was released on 2003-03-31. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

We would like to thank the OpenSSH community for their continued
support to the project, especially those who contributed source and
bought T-shirts or posters.

We have a new design of T-shirt available, more info on
        http://www.openbsd.org/tshirts.html#18

For international orders use http://https.openbsd.org/cgi-bin/order
and for European orders, use http://https.openbsd.org/cgi-bin/order.eu

Changes since OpenSSH 3.5:
==========

* RSA blinding is now used by ssh(1), sshd(8) and ssh-agent(1).

in order to avoid potential timing attacks against the RSA keys.
Older versions of OpenSSH have been using RSA blinding in
ssh-keysign(1) only.

Please note that there is no evidence that the SSH protocol is
vulnerable to the OpenSSL/TLS timing attack described in
http://crypto.stanford.edu/~dabo/papers/ssl-timing.pdf

* ssh-agent(1) optionally requires user confirmation if a key gets
used, see '-c' in ssh-add(1).

* sshd(8) now handles PermitRootLogin correctly when UsePrivilegeSeparation
is enabled.

* sshd(8) now removes X11 cookies when a session gets closed.

* ssh-keysign(8) is disabled by default and only enabled if the
new EnableSSHKeysign option is set in the global ssh_config(5)
file.

* ssh(1) and sshd(8) now handle 'kex guesses' correctly (key exchange
guesses).

* ssh(1) no longer overwrites SIG_IGN. This matches behaviour from
rsh(1) and is used by backup tools.

* setting ProxyCommand to 'none' disables the proxy feature, see
ssh_config(5).

* scp(1) supports add -1 and -2.

* scp(1) supports bandwidth limiting.

* sftp(1) displays a progressmeter.

* sftp(1) has improved error handling for scripting.

Checksums:
==========

- MD5 (openssh-3.6p1.tar.gz) = 72ef1134d521cb6926c99256dad17fe0
- MD5 (openssh-3.6.tgz) = 758822b888c5c3f83a98045aef904254

Reporting Bugs:
===============

- please read http://www.openssh.com/report.html
and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller and Ben Lindstrom.

OpenSSH 3.5/3.5p1 (2002-10-15)

OpenSSH 3.5 was released on 2002-10-15. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

We would like to thank the OpenSSH community for their continued
support and encouragement.

Changes since OpenSSH 3.4:
==========================

* Improved support for Privilege Separation (Portability, Kerberos,
PermitRootLogin handling).

* ssh(1) prints out all known host keys for a host if it receives an
unknown host key of a different type.

* Fixed AES/Rijndael EVP integration for OpenSSL < 0.9.7 (caused
problems with bounds checking patches for gcc).

* ssh-keysign(8) is disabled by default and only enabled if the
HostbasedAuthentication option is enabled in the global ssh_config(5)
file.

* ssh-keysign(8) uses RSA blinding in order to avoid timing attacks
against the RSA host key.

* A use-after-free bug was fixed in ssh-keysign(8). This bug
broke hostbased authentication on several platforms.

* ssh-agent(1) is now installed setgid in order to avoid ptrace(2)
attacks.

* ssh-agent(1) now restricts the access with getpeereid(2) (or
equivalent, where available).

* sshd(8) no longer uses the ASN.1 parsing code from libcrypto when
verifying RSA signatures.

```
* sshd(8) now sets the SSH_CONNECTION environment variable.

* Enhanced "ls" support for the sftp(1) client, including globbing and
  detailed listings.

* ssh(1) now always falls back to uncompressed sessions, if the
  server does not support compression.

* The default behavior of sshd(8) with regard to user settable
  environ variables has changed:  the new option PermitUserEnvironment
  is disabled by default, see sshd_config(5).

* The default value for LoginGraceTime has been changed from 600 to 120
  seconds, see sshd_config(5).

* Removed erroneous SO_LINGER handling.


Checksums:
==========

- MD5 (openssh-3.5p1.tar.gz) = 42bd78508d208b55843c84dd54dea848
- MD5 (openssh-3.5.tgz) = 79fc225dbe0fe71ebb6910f449101d23


Reporting Bugs:
===============

- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller and Ben Lindstrom.
```

OpenSSH 3.4/3.4p1 (2002-06-26)

```
OpenSSH 3.4 was released on 2002-06-26. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

We would like to thank the OpenSSH community for their continued
support and encouragement.


Changes since OpenSSH 3.3:
==========================

Security Changes:
=================

  All versions of OpenSSH's sshd between 2.9.9 and 3.3
  contain an input validation error that can result in
  an integer overflow and privilege escalation.

  OpenSSH 3.4 fixes this bug.

  In addition, OpenSSH 3.4 adds many checks to detect
  invalid input and mitigate resource exhaustion attacks.

  OpenSSH 3.2 and later prevent privilege escalation
  if UsePrivilegeSeparation is enabled in sshd_config.
  OpenSSH 3.3 enables UsePrivilegeSeparation by
  default.


Reporting Bugs:
===============

- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller and Ben Lindstrom.
```

OpenSSH 3.3/3.3p1 (2002-06-21)

```
OpenSSH 3.3 was released on 2002-06-21. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

We would like to thank the OpenSSH community for their continued
support and encouragement.


Changes since OpenSSH 3.2.3:
============================

Security Changes:
=================

- improved support for privilege separation:
```

```
        privilege separation is now enabled by default

  See UsePrivilegeSeparation in sshd_config(5)
  and http://www.citi.umich.edu/u/provos/ssh/privsep.html for more
  information.
- ssh no longer needs to be installed setuid root for protocol
  version 2 hostbased authentication, see ssh-keysign(8).
  protocol version 1 rhosts-rsa authentication still requires privileges
  and is not recommended.

Other Changes:
==============

- documentation for the client and server configuration options have
  been moved to ssh_config(5) and sshd_config(5).
- the server now supports the Compression option, see sshd_config(5).
- the client options RhostsRSAAuthentication and RhostsAuthentication now
  default to no, see ssh_config(5).
- the client options FallBackToRsh and UseRsh are deprecated.
- ssh-agent now supports locking and timeouts for keys, see ssh-add(1).
- ssh-agent can now bind to unix-domain sockets given on the command line,
  see ssh-agent(1).
- fixes problems with valid RSA signatures from putty clients.

Reporting Bugs:
===============

- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller and Ben Lindstrom.
```

OpenSSH 3.2.3/3.2.3p1 (2002-05-23)

```
OpenSSH 3.2.3 was released on 2002-05-23. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

We would like to thank the OpenSSH community for their continued
support and encouragement.


Changes since OpenSSH 3.2.2:
============================

  This release fixes several problems in OpenSSH 3.2.2:

    - a defect in the BSD_AUTH access control handling for
      OpenBSD and BSD/OS systems:

      Under certain conditions, on systems using YP with netgroups
      in the password database, it is possible that sshd does ACL
      checks for the requested user name but uses the password
      database entry of a different user for authentication. This
      means that denied users might authenticate successfully while
      permitted users could be locked out (OpenBSD PR 2659).

    - login/tty problems on Solaris (bug #245)

    - build problems on Cygwin systems


Changes between OpenSSH 3.1 and OpenSSH 3.2.2:
==============================================

  Security Changes:
  =================

  - fixed buffer overflow in Kerberos/AFS token passing
  - fixed overflow in Kerberos client code
  - sshd no longer auto-enables Kerberos/AFS
  - experimental support for privilege separation,
    see UsePrivilegeSeparation in sshd(8) and
          http://www.citi.umich.edu/u/provos/ssh/privsep.html
    for more information.
  - only accept RSA keys of size SSH_RSA_MINIMUM_MODULUS_SIZE (768) or larger

  Other Changes:
  ==============

  - improved smartcard support (including support for OpenSC,
    see www.opensc.org)
  - improved Kerberos support (including support for MIT-Kerberos V)
  - fixed stderr handling in protocol v2
  - client reports failure if -R style TCP forwarding fails in protocol v2
  - support configuration of TCP forwarding during interactive sessions (~C)
  - improved support for older sftp servers
  - improved support for importing old DSA keys (from ssh.com software).
  - client side suport for PASSWD_CHANGEREQ in protocol v2
  - fixed waitpid race conditions
  - record correct lastlogin time

Reporting Bugs:
===============
```

```
- please read http://www.openssh.com/report.html
  and http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller and Ben Lindstrom.
```

OpenSSH 3.2.2/3.2.2p1 (2002-05-16)

```
OpenSSH 3.2.2 was released on 2002-05-16. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

We would like to thank the OpenSSH community for their continued
support and encouragement.

Security Changes:
=================

- fixed buffer overflow in Kerberos/AFS token passing
- fixed overflow in Kerberos client code
- sshd no longer auto-enables Kerberos/AFS
- experimental support for privilege separation,
  see UsePrivilegeSeparation in sshd(8) and
          http://www.citi.umich.edu/u/provos/ssh/privsep.html
  for more information.
- only accept RSA keys of size SSH_RSA_MINIMUM_MODULUS_SIZE (768) or larger

Other Changes:
==============

- improved smartcard support (including support for OpenSC, see www.opensc.org)
- improved Kerberos support (including support for MIT-Kerberos V)
- fixed stderr handling in protocol v2
- client reports failure if -R style TCP forwarding fails in protocol v2
- support configuration of TCP forwarding during interactive sessions (~C)
- improved support for older sftp servers
- improved support for importing old DSA keys (from ssh.com software).
- client side suport for PASSWD_CHANGEREQ in protocol v2
- fixed waitpid race conditions
- record correct lastlogin time

Reporting Bugs:
===============

- please read http://www.openssh.com/report.html and
  http://bugzilla.mindrot.org/

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller and Ben Lindstrom.
```

OpenSSH 3.1/3.1p1 (2004-04-09)

```
OpenSSH 3.1 was released on 2004-04-09. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

We would like to thank the OpenSSH community for their continued
support and encouragement.


Important Changes:
==================

- /etc/ssh/ now default directory for keys and configuration files
- ssh-keygen no longer defaults to a specific key type (rsa1);
  use ssh-keygen -t {rsa,dsa,rsa1}
- sshd x11 forwarding listens on localhost by default;
  see sshd X11UseLocalhost option to revert to prior behaviour
  if your older X11 clients do not function with this configuration


Other Changes:
==============

- ssh ~& escape char functions now for both protocol versions
- sshd ReverseMappingCheck option changed to VerifyReverseMapping
  to clarify its function; ReverseMappingCheck can still be used
- public key fingerprint is now logged with LogLevel=VERBOSE
- reason logged for disallowed logins (e.g., no shell, etc.)
- more robust error handling for x11 forwarding
- improved packet/window size handling in ssh2
- use of regex(3) has been removed
- fix SIGCHLD races in sshd (seen on Solaris)
- sshd -o option added
- sftp -B -R -P options added
- ssh-add now adds all 3 default keys
- ssh-keyscan bug fixes
- ssh-askpass for hostkey dialog
- fix fd leak in sshd on SIGHUP
- TCP_NODELAY set on X11 and TCP forwarding endpoints
```

```
OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller and Ben Lindstrom.
```

OpenSSH 3.0.2/3.0.2p1 (2002-12-04)

```
OpenSSH 3.0.2 was released on 2002-12-04. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

We would like to thank the OpenSSH community for their continued
support and encouragement.

Important Changes:
==================

        This release fixes a vulnerability in the UseLogin option
        of OpenSSH.  This option is not enabled in the default
        installation of OpenSSH.

        However, if UseLogin is enabled by the administrator, all
        versions of OpenSSH prior to 3.0.2 may be vulnerable to
        local attacks.

        The vulnerability allows local users to pass environment
        variables (e.g. LD_PRELOAD) to the login process.  The login
        process is run with the same privilege as sshd (usually
        with root privilege).

        Do not enable UseLogin on your machines or disable UseLogin
        again in /etc/sshd_config:
                UseLogin no

We also have received many reports about attacks against the crc32
bug.  This bug has been fixed about 12 months ago in OpenSSH 2.3.0.
However, these attacks cause non-vulnerable daemons to chew a lot
of cpu since the crc32 attack sends a tremendously large amount of
data which must be processed.

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller and Ben Lindstrom.



The following patch fixes the UseLogin vulnerability in OpenSSH 3.0.1 and
earlier releases.

--- session.c   11 Oct 2001 13:45:21 -0000      1.108
+++ session.c   1 Dec 2001 22:14:39 -0000
@@ -875,6 +875,7 @@
                child_set_env(&env, &envsize, "TZ", getenv("TZ"));

        /* Set custom environment options from RSA authentication. */
+       if (!options.use_login)
        while (custom_environment) {
                struct envstring *ce = custom_environment;
                char *s = ce->s;
```

OpenSSH 3.0.1/3.0.1p1 (2001-11-19)

```
OpenSSH 3.0.1 was released on 2001-11-19. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

We would like to thank the OpenSSH community for their continued
support and encouragement.

Important Changes:
==================

        A security hole that may allow an attacker to authenticate
        if -- and only if -- the administrator has enabled KerberosV.
        By default, OpenSSH KerberosV support only becomes active
        after KerberosV has been properly configured.

        An excessive memory clearing bug (which we believe to be
        unexploitable) also exists, but since this may cause daemon
        crashes, we are providing a fix as well.

        Various other non-critical fixes (~& support and more).

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller and Ben Lindstrom.
```

OpenSSH 3.0/3.0p1 (2001-11-06)

```
OpenSSH 3.0 was released on 2001-11-06. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.
```

```
This release contains many portability bug-fixes (listed in the
ChangeLog) as well as several new features (listed below).

We would like to thank the OpenSSH community for their continued
support and encouragement.

Important Changes:
==================

1) SSH protocol v2 is now the default protocol version

        use the 'Protocol' option from ssh(1) and sshd(8) if
        you need to change this.

2) The files
        /etc/ssh_known_hosts2
        ~/.ssh/known_hosts2
        ~/.ssh/authorized_keys2
   are now obsolete, you can use
        /etc/ssh_known_hosts
        ~/.ssh/known_hosts
        ~/.ssh/authorized_keys
   For backward compatibility ~/.ssh/authorized_keys2 will still used for
   authentication and hostkeys are still read from the known_hosts2.
   However, those deprecated files are considered 'readonly'.  Future
   releases are likely not to read these files.

3) The CheckMail option in sshd_config is deprecated, as sshd(8) no longer
   checks for new mail.

4) X11 cookies are now stored in $HOME.

New Features:
=============

1) Smartcard support in the ssh client and agent based on work by
   University of Michigan CITI (http://www.citi.umich.edu/projects/smartcard/).
2) support for Rekeying in protocol version 2

3) improved Kerberos support in protocol v1 (KerbIV and KerbV)

4) backward compatibility with older commercial SSH versions >= 2.0.10

5) getopt(3) is now used by all programs

6) dynamic forwarding (use ssh(1) as your socks server)

7) ClearAllForwardings in ssh(1)

8) ssh(1) now checks the hostkey for localhost (NoHostAuthenticationForLocalhost yes/no).

9) -F option in ssh(1)

10) ssh(1) now has a '-b bindaddress' option

11) scp(1) allows "scp /file localhost:/file"

12) The AuthorizedKeysFile option allows specification of alternative
    files that contain the public keys that can be used for user authentication
    (e.g. /etc/ssh_keys/%u, see sshd(8))

13) extended AllowUsers user@host syntax in sshd(8)

14) improved challenge-response support (especially for systems supporting BSD_AUTH)

15) sshd(8) can specify time args as 1h, 2h30s etc.

16) sshd(8) transmits the correct exit status for remote execution with protocol version 2.

17) ssh-keygen(1) can import private RSA/DSA keys generated with the commercial version

18) ssh-keyscan(1) supports protocol version 2

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller and Ben Lindstrom.
```

OpenSSH 2.9p2 (2001-06-17)

```
Portable OpenSSH 2.9p2 has just been uploaded and shall be making its
way to the mirrors listed at http://www.openssh.com/portable.html
shortly.

This release fixes the "cookies" file deletion problem reported on
BUGTRAQ as well as a few other minor (non-security) bugs. No new
features have been added in this release.

Regards,
Damien Miller
```

OpenSSH 2.9.9/2.9.9p1 (2001-09-25)

```
OpenSSH 2.9.9 has just been uploaded. It is available from the
mirrors listed at https://www.openssh.com/.
```

```
OpenSSH 2.9.9 fixes a weakness in the key file option handling,
including source IP based access control.

OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

This release contains many portability bug-fixes (listed in the
ChangeLog) as well as several new features (listed below).

We would like to thank the OpenSSH community for their continued
support and encouragement.

Security Notes:
===============

This release fixes weakness in the source IP based access control
for SSH protocol v2 public key authentication:

        Versions of OpenSSH between 2.5 and 2.9.9 are
        affected if they use the 'from=' key file option in
        combination with both RSA and DSA keys in
        ~/.ssh/authorized_keys2.

        Depending on the order of the user keys in
        ~/.ssh/authorized_keys2 sshd might fail to apply the
        source IP based access control restriction (e.g.
        from="10.0.0.1") to the correct key:

        If a source IP restricted key (e.g. DSA key) is
        immediately followed by a key of a different type
        (e.g. RSA key), then key options for the second key
        are applied to both keys, which includes 'from='.

        This means that users can circumvent the system policy
        and login from disallowed source IP addresses.


Important Changes:
==================

OpenSSH 2.9.9 might have upgrade issues introduced by the long time
between releases, which may affect people in unforseen ways:

1) The files
        /etc/ssh_known_hosts2
        ~/.ssh/known_hosts2
        ~/.ssh/authorized_keys2
   are now obsolete, you can use
        /etc/ssh_known_hosts
        ~/.ssh/known_hosts
        ~/.ssh/authorized_keys
   For backward compatibility ~/.ssh/authorized_keys2 is still used for
   authentication and hostkeys are still read from the known_hosts2.
   However, old files are considered 'readonly'.  Future releases are
   likely to not read these files.

2) The CheckMail option in sshd_config is deprecated, sshd no longer
   checks for new mail.

3) X11 cookies are stored in $HOME

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller and Ben Lindstrom.
```

OpenSSH 2.9/2.9p1 (2001-04-29)

```
OpenSSH 2.9 has just been uploaded. It is available from the
mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol version 1.3, 1.5 and 2.0
implementation and includes sftp client and server support.

This release contains many portability bug-fixes (listed in the
ChangeLog) as well as several new features (listed below).

We would like to thank the OpenSSH community for their continued
support and encouragement.

Important Changes:
==================

WARNING: SSH protocol v2 is now the default protocol version

        use the 'Protocol' option from ssh(1) and sshd(8) if
        you want to change this.

SSH protocol v2 implementation adds support for:

        HostbasedAuthentication, similar to RhostsRSA in SSH protocol
        v1

        Rekeying (negotiate new encryption keys for the current SSH
        session, try ~R in interactive SSH sessions)

        updated DH group exchange:
                draft-ietf-secsh-dh-group-exchange-01.txt
```

```
        client option HostKeyAlgorithms

        server options ClientAliveInterval and ClientAliveCountMax

        tty mode passing

general:

        gid swapping in sshd (fixes access to /home/group/user based
        directory structures)

        Dan Kaminsky <dankamin@cisco.com> contributed an experimental
        SOCKS4 proxy to the ssh client (yes, client not the server).
        Use 'ssh -D 1080 server' if you want to try this out.

        server option PrintLastLog

        improvements for scp > 2GB

        improved ListenAddress option.
        You can now use ListenAddress host:port

        improved interoperability (bug detection for older implementations)

        improved documentation

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller and Ben Lindstrom.
```

OpenSSH 2.5.2p2 (2001-03-22)

```
Portable OpenSSH 2.5.2p2 is now available from the mirror sites
listed at http://www.openssh.com/portable.html

Security related changes:
        Improved countermeasure against "Passive Analysis of SSH
        (Secure Shell) Traffic"
        http://openwall.com/advisories/OW-003-ssh-traffic-analysis.txt

        The countermeasures introduced in earlier OpenSSH-2.5.x versions
        caused interoperability problems with some other implementations.

        Improved countermeasure against "SSH protocol 1.5 session
        key recovery vulnerability"
        http://www.core-sdi.com/advisories/ssh1_sessionkey_recovery.htm

New options:
        permitopen authorized_keys option to restrict portforwarding.

        PreferredAuthentications allows client to specify the order in which
        authentication methods are tried.

Sftp:
        sftp client supports globbing (get *, put *).

        Support for sftp protocol v3 (draft-ietf-secsh-filexfer-01.txt).

        Batch file (-b) support for automated transfers

Performance:
        Speedup DH exchange. OpenSSH should now be significantly faster when
        connecting use SSH protocol 2.

        Preferred SSH protocol 2 cipher is AES with hmac-md5. AES offers
        much faster throughput in a well scrutinised cipher.

Bugfixes:
        stderr handling fixes in SSH protocol 2.

        Improved interoperability.

Client:
        The client no longer asks for the the passphrase if the key
        will not be accepted by the server (SSH2_MSG_USERAUTH_PK_OK)

Miscellaneous:
        scp should now work for files > 2GB

        ssh-keygen can now generate fingerprints in the "bubble babble"
        format for exchanging fingerprints with SSH.COM's SSH protocol 2
        implementation.

Portable version:
        Better support for the PRNGd[1] entropy collection daemon. The
        --with-egd-pool configure option has been deprecated in favour
        of --with-prngd-socket and the new --with-prngd-port options.
        The latter allows collection of entropy from a localhost
        socket.

        configure ensures that scp is in the $PATH set by the server
        (unless a custom path is specified).

-d

[1] http://www.aet.tu-cottbus.de/personen/jaenicke/postfix_tls/prngd.html
```

OpenSSH 2.5.1p2 (2001-03-20)

```
Portable OpenSSH 2.5.1p2 has just been uploaded and will be making its
way to the mirror sites (http://www.openssh.com/portable.html)in due
course.

This release contains primarily bug-fixes over 2.5.1p1 but an upgrade is
recommended. Specific bug-fixes include:

 - Fixed endianess issue causing failues when usin Rijndael/AES cipher
 - Fix PAM failures on Solaris and Linux
 - Fix RPM spec file for Redhat systems
 - Fixed several compatibility functions
 - Fix entropy collection code for SCO3 and NeXTStep
 - Many other minor fixes (see Changelog for details)

This release includes Mark Roth's mdoc2man.pl script which can be used
to fix up the manpages on systems that lack the full andoc set of
macros (e.g. Solaris). A future release of portable OpenSSH will automate
this scripts use for systems that require it.

-d
```

OpenSSH 2.5.1p1 (2001-02-19)

```
Portable OpenSSH 2.5.1p1 has just been uploaded. It will be available
from the mirrors listed at http://www.openssh.com/portable.html shortly.

OpenSSH is a 100% complete SSH 1.3 & 1.5 protocol implementation and
a 99% SSH 2 protocol implementation, including sftp client and server
support.

This release contains many portability bug-fixes (listed in the
ChangeLog) as well as several new features (listed below).

OpenSSH 2.5.0p1 was skipped because of interoperability issues with
ssh-1.2.18 => ssh-1.2.22.

We would like to thank the OpenSSH community for their continued support
and encouragement.

Important Changes:
==================

1) Features added to the implementation of the SSH 2 protocol:

    * agent forwarding
    * support for -R forwarding
    * RSA host and userkeys
    * extended support for older SSH 2 protocol implementations

    OpenSSH still lacks support for rekeying, so you have to turn off
    rekeying if your server tries to force this feature.

    The next release of OpenSSH will probably support rekeying.

2) Damien Miller contributed an interactive sftp client.

    The sftp client works for both SSH protocol versions.

3) David Mazieres' ssh-keyscan has been added to the OpenSSH distribution.

4) Now there are three types of keys in OpenSSH:

    RSA1 is used by the SSH 1 protocol only,
    RSA and DSA keys are used by the SSH 2 protocol implementation.

    You can generate RSA keys for use with SSH 2 protocol with:

        $ ssh-keygen -t rsa -f /etc/ssh_host_rsa_key

        To use RSA or DSA keys in SSH 2 protocol, simply
        add the public keys to the .ssh/authorised_keys2 file.

    IdentityFile2, HostDsaKey and DSAAuthentication are obsolete:

    You can use multiple IdentityFile and HostKey options instead, e.g
        HostKey /etc/ssh_host_key
        HostKey /etc/ssh_host_dsa_key
        HostKey /etc/ssh_host_rsa_key
    in /etc/sshd_config

    The option DSAAuthentication has been replaced by PubkeyAuthentication.

    Fingerprinting works for all types of keys:

        $ ssh-keygen -l -f $HOME/.ssh/{authorized_keys,known_hosts}{,2}

5) Important changes in the implementation of SSH 1 protocol:

    The OpenSSH server does not require a privileged source port for
    RhostsRsaAuthentication, since it adds no additional security.

    Interoperation with SSH 1.4 protocol
```

```
6) New option HostKeyAlias

    This option allows the user to record the host key under a
    different name. This is useful for tunneling over
    forwarded connections or if you run multiple sshd's on
    different ports on the same machine.

    Alternatively you can use the UserKnownHostsFile or
    UserKnownHostsFile2 options to specify separate host key
    files for the connection.

7) The ReverseMappingCheck is now optional in sshd_config.

    If you combine this with the 'sshd -u0' option the server
    will not do DNS lookups when a client connects.

8) Stricter Hostkey Checking

9) Option Change Summary:

    a) New or changed:

        ChallengeResponseAuthentication
        MACs
        PubkeyAuthentication

        HostkeyAlias        (Client only)

        Banner              (Server only)
        ReverseMappingCheck (Server only)

        PermitRootLogin     {yes,without-password,forced-commands-only,no}

        {Allow,Deny}Groups  now support supplementary groups

        sshd -D             for monitoring scripts or inittab
        ssh -t              multiple -t force tty allocation

    b) Obsolete:

        DsaAuthentication   (use PubkeyAuthentication instead)
        HostDsaKey          (use HostKey)
        Identityfile2       (use Identityfile or -i)
        SkeyAuthentication  (use ChallengeResponseAuthentication)
        TisAuthentication   (use ChallengeResponseAuthentication)

OpenSSH is brought to you by Markus Friedl, Niels Provos, Theo de Raadt,
Kevin Steves, Damien Miller and Ben Lindstrom.
```

OpenSSH 2.3.0p1 (2000-11-06)

```
This is to announce the release of portable openssh-2.3.0p1. This
release includes many new features and bug fixes. This is a
recommended upgrade if you are using 2.2.0p1 or an older release.

Portable OpenSSH is available from one of the many mirrors listed at
http://www.openssh.com/portable.html

Some of the more notable features include:

- Rijndael support for SSH2. Use the "Ciphers" configuration directive
  to enable it. (Markus Friedl <markus@cvs.openbsd.org>

- Cygwin support (Corinna Vinschen <vinschen@cygnus.com>)

- sftp-server support (Markus Friedl <markus@cvs.openbsd.org>)

- SSH1 single-des support for interop with Cisco routers. This cipher
  is never enabled automatically, you have to either specify "-c des"
  or enable it using "Cipher des" in a config file. (Markus Friedl
  <markus@cvs.openbsd.org>

- Support expired password change through PAM (Steve VanDevender's
  <stevev@darkwing.uoregon.edu>)

- Better compatibility with buggy SSH implementations (Markus Friedl
  <markus@cvs.openbsd.org>

- S/key support for SSH2, based on kbd-interactive auth (Markus Friedl
  <markus@cvs.openbsd.org> and mkiernan@avantgo.com

- scp now supports "-o" option (Ben Lindstron <mouring@pconline.com>)

Please refer to the ChangeLog for a full list of features and bugfixes.

Regards,
Damien Miller
```

OpenSSH 2.2.0p1 (2000-09-01)

```
Version 2.2.0p1 of portable OpenSSH has just been uploaded to the
master site and should be making its way to the mirrors in due
course.
```

http://www.openssh.com/portable.html

This release contains several new features and bugfixes relative to
the previous 2.1.1p4 release. In particular:

- DSA key support in ssh-agent. Please not that this will not
  interop with ssh.com's ssh-agent (Markus Friedl)
- sshd now implements Random Early Drop connection rate limiting,
  which can help mitigate DoS attacks against sshd. See the
  `MaxStartups' option in the sshd manpage for details (Markus Friedl)
- `-u' option to sshd allow logging of hostnames (rather than IP
  addresses) in wtmp when `UseLogin' is set to `yes'. (Markus Friedl)
- Escape character `~' support in SSH2 (Markus Friedl)
- Interop with SSH.COM ssh 2.3.0 (Markus Friedl)
- Fix problems when sshd is run from inetd
- Better SunOS 4.1.x support (Nate Itkin and Charles Levert)
- Solaris package support, see contrib/solaris (Rip Loomis)
- Work around connection freezes on HPUX and SunOS 4 (Lutz Jaenicke,
  Tamito KAJIYAMA)
- Fix ^C ignored issue on Solaris. (Gert Doering, John Horne and
  Garrick James)
- Further improved NeXT support. (Ben Lindstrom, Mark Miller)
- Lots of other minor fixes (see ChangeLog for details)

This release has been tested on HPUX (10.20, 11.00), Irix (5.3,
6.5), Linux (Debian, Redhat, Slackware, SuSE), NeXTstep 3 (HPPA,
i386, m68k), OpenStep (i386, m68k, Sparc), SCO Unixware 7.1.0, SCO
OpenServer 5.0.5, Solaris 2.7 (Sparc), Solaris 2.8 (i386, Sparc),
SNI/Reliant Unix, DEC OSF/Tru64 5.0.

Many thanks to those who contributed bug reports, fixes and testing
time.

Regards,
Damien Miller

OpenSSH 2.1.1p4 (2000-07-16)

I have just uploaded portable OpenSSH 2.1.1p4, it should be making
its way to the mirrors listed at http://www.openssh.com/portable.html
soon.

This release contains several bugfixes from the OpenBSD team,
primarily the config file parsing problem reported by Ralf
Engelschall <rse@engelschall.com>

Regards,
Damien Miller

--------------- Changelog:

```
20000716
 - Release 2.1.1p4

20000715
 - (djm) OpenBSD CVS updates
   - provos@cvs.openbsd.org  2000/07/13 16:53:22
     [aux.c readconf.c servconf.c ssh.h]
     allow multiple whitespace but only one '=' between tokens, bug report from
     Ralf S. Engelschall <rse@engelschall.com> but different fix. okay deraadt@
   - provos@cvs.openbsd.org  2000/07/13 17:14:09
     [clientloop.c]
     typo; todd@fries.net
   - provos@cvs.openbsd.org  2000/07/13 17:19:31
     [scp.c]
     close can fail on AFS, report error; from Greg Hudson <ghudson@mit.edu>
   - markus@cvs.openbsd.org  2000/07/14 16:59:46
     [readconf.c servconf.c]
     allow leading whitespace. ok niels
   - djm@cvs.openbsd.org     2000/07/14 22:01:38
     [ssh-keygen.c ssh.c]
     Always create ~/.ssh with mode 700; ok Markus
 - Fixes for SunOS 4.1.4 from Gordon Atwood <gordon@cs.ualberta.ca>
   - Include floatingpoint.h for entropy.c
   - strerror replacement

---------------
```

OpenSSH 2.1.1p3 (2000-07-12)

The 2.1.1p3 release of portable OpenSSH has been uploaded to the
OpenBSD ftp master site. In a few hours it will be available from one
of the many mirrors listed at:

http://www.openssh.com/portable.html

This release fixes several bugs reported since the previous release
and extends portability to NeXT and Reliant Unix.

As usual, the OpenBSD team has been hard at work further polishing and
enhancing OpenSSH. This release brings a new configuration directive
"MaxStartups" which mitigates connection flooding attacks, further

```
details are in the sshd man-page.

Another noteworthy difference from previous releases is that
'FallBackToRsh' now defaults to 'no'. Users of this feature may need
to edit their /etc/ssh_config or ~/.ssh/config files to achieve the
same behavior.

Again, thanks to those who reported bugs, tested the snapshot and sent
fixes.

Regards,
Damien Miller

------------------ Changelog

20000712
 - (djm) Remove -lresolve for Reliant Unix
 - (djm) OpenBSD CVS Updates:
   - deraadt@cvs.openbsd.org 2000/07/11 02:11:34
     [session.c sshd.c ]
     make MaxStartups code still work with -d; djm
   - deraadt@cvs.openbsd.org 2000/07/11 13:17:45
     [readconf.c ssh_config]
     disable FallBackToRsh by default
 - (djm) Replace in_addr_t with u_int32_t in bsd-inet_aton.c. Report from
   Ben Lindstrom <mouring@pconline.com>
 - (djm) Make building of X11-Askpass and GNOME-Askpass optional in RPM
   spec file.
 - (djm) Released 2.1.1p3

20000711
 - (djm) Fixup for AIX getuserattr() support from Tom Bertelson
   <tbert@abac.com>
 - (djm) ReliantUNIX support from Udo Schweigert <ust@cert.siemens.de>
 - (djm) NeXT: dirent structures to get scp working from Ben Lindstrom
   <mouring@pconline.com>
 - (djm) Fix broken inet_ntoa check and ut_user/ut_name confusion, report
   from Jim Watt <jimw@peisj.pebio.com>
 - (djm) Replaced bsd-snprintf.c with one from Mutt source tree, it is known
   to compile on more platforms (incl NeXT).
 - (djm) Added bsd-inet_aton and configure support for NeXT
 - (djm) Misc NeXT fixes from Ben Lindstrom <mouring@pconline.com>
 - (djm) OpenBSD CVS updates:
   - markus@cvs.openbsd.org  2000/06/26 03:22:29
     [authfd.c]
     cleanup, less cut&paste
   - markus@cvs.openbsd.org  2000/06/26 15:59:19
     [servconf.c servconf.h session.c sshd.8 sshd.c]
     MaxStartups: limit number of unauthenticated connections, work by
     theo and me
   - deraadt@cvs.openbsd.org 2000/07/05 14:18:07
     [session.c]
     use no_x11_forwarding_flag correctly; provos ok
   - provos@cvs.openbsd.org  2000/07/05 15:35:57
     [sshd.c]
     typo
   - aaron@cvs.openbsd.org   2000/07/05 22:06:58
     [scp.1 ssh-agent.1 ssh-keygen.1 sshd.8]
     Insert more missing .El directives. Our troff really should identify
     these and spit out a warning.
   - todd@cvs.openbsd.org    2000/07/06 21:55:04
     [auth-rsa.c auth2.c ssh-keygen.c]
     clean code is good code
   - deraadt@cvs.openbsd.org 2000/07/07 02:14:29
     [serverloop.c]
     sense of port forwarding flag test was backwards
   - provos@cvs.openbsd.org  2000/07/08 17:17:31
     [compat.c readconf.c]
     replace strtok with strsep; from David Young <dyoung@onthejob.net>
   - deraadt@cvs.openbsd.org 2000/07/08 19:21:15
     [auth.h]
     KNF
   - ho@cvs.openbsd.org      2000/07/08 19:27:33
     [compat.c readconf.c]
     Better conditions for strsep() ending.
   - ho@cvs.openbsd.org      2000/07/10 10:27:05
     [readconf.c]
     Get the correct message on errors. (niels@ ok)
   - ho@cvs.openbsd.org      2000/07/10 10:30:25
     [cipher.c kex.c servconf.c]
     strtok() --> strsep(). (niels@ ok)
 - (djm) Fix problem with debug mode and MaxStartups
 - (djm) Don't generate host keys when $(DESTDIR) is set (e.g. during RPM
   builds)
 - (djm) Add strsep function from OpenBSD libc for systems that lack it

20000709
 - (djm) Only enable PAM_TTY kludge for Linux. Problem report from
   Kevin Steves <stevesk@sweden.hp.com>
 - (djm) Match prototype and function declaration for rresvport_af.
   Problem report from Niklas Edmundsson <nikke@ing.umu.se>
 - (djm) Missing $(DESTDIR) on host-key target causing problems with RPM
   builds. Problem report from Gregory Leblanc <GLeblanc@cu-portland.edu>
 - (djm) Replace ut_name with ut_user. Patch from Jim Watt
   <jimw@peisj.pebio.com>
 - (djm) Fix pam sprintf fix
 - (djm) Cleanup entropy collection code a little more. Split initialisation
   from seeding, perform intialisation immediatly at start, be careful with
```

```
    uids. Based on problem report from Jim Watt <jimw@peisj.pebio.com>
  - (djm) More NeXT compatibility from Ben Lindstrom <mouring@pconline.com>
    Including sigaction() et al. replacements
  - (djm) AIX getuserattr() session initialisation from Tom Bertelson
    <tbert@abac.com>

20000708
  - (djm) Fix bad fprintf format handling in auth-pam.c. Patch from
    Aaron Hopkins <aaron@die.net>
  - (djm) Fix incorrect configure handling of --with-rsh-path option. Fix from
    Lutz Jaenicke <Lutz.Jaenicke@aet.TU-Cottbus.DE>
  - (djm) Fixed undefined variables for OSF SIA. Report from
    Baars, Henk <Hendrik.Baars@nl.origin-it.com>
  - (djm) Handle EWOULDBLOCK returns from read() and write() in atomicio.c
    Fix from Marquess, Steve Mr JMLFDC <Steve.Marquess@DET.AMEDD.ARMY.MIL>
  - (djm) Don't use inet_addr.

20000702
  - (djm) Fix brace mismatch from Corinna Vinschen <vinschen@cygnus.com>
  - (djm) Stop shadow expiry checking from preventing logins with NIS. Based
    on fix from HARUYAMA Seigo <haruyama@nt.phys.s.u-tokyo.ac.jp>
  - (djm) Use standard OpenSSL functions in auth-skey.c. Patch from
    Chris, the Young One <cky@pobox.com>
  - (djm) Fix scp progress meter on really wide terminals. Based on patch
    from James H. Cloos Jr. <cloos@jhcloos.com>

------------------
```

OpenSSH 2.1.1p2 (2000-07-01)

```
Announcing the release of portable OpenSSH 2.1.1p2.

This release primarily contains fixes to the bugs that have been
reported over the last month, in particular:

 - Invalid time bring written to utmp/wtmp on systems using bash2

 - Several lastlog fixes

 - AIX, SCO, Irix portability fixes

 - Avoid failures on PAM systems when using PAM authentication modules
   which require a tty.

 - Entropy collection fixes for Solaris.

 - EGD robustness improvements

 - Fixes and enhancements from the OpenBSD team:
   - Fixed options processing in authorized_keys2 file
   - Compatibility with commercial SSH 2.0.13 and 2.2.0
   - Numerous minor fixes

There are also a couple of new features:

 - Shadow password expiry support (no password change support yet)

 - Irix 6.x array sessions, project IDs and system audit trail IDs

 - Beginnings of Tru64 / OSF SIA (Security Integration Architecture)
   support

 - Beginnings of NeXT support

Version 2.1.1p2 will be available from the mirrors listed at
http://www.openssh.com/portable.html (as soon as they update).

Many thanks to all those who tested the snapshots and/or contributed
bug reports and patches

Regards,
Damien Miller
```

OpenSSH 2.1.1p1 (2000-06-09)

```
Announcing the availability of portable OpenSSH 2.1.1p1.

This release contains the fix for the "UseLogin yes" vulnerability
identified in Markus' release and several other enhancements and
bugfixes. Including:

 - Better login code. Andre Lucas has rewritten the login code to
   be much more modular and extensible. In the process he has fixed
   the problems with Solaris utmp[x].

 - Revised the entropy collection code to be faster and more reliable.

 - Fix for RSA host restrictions ("from=" in authorized_keys)

It is recommended that all users upgrade to this version.

Portable OpenSSH 2.1.1p1 is available from one of the many mirrors
listed at: http://www.openssh.com/portable.html
```

```
Regards,
Damien Miller
```

OpenSSH 2.1.0p1 (2000-05-09)

```
This is to announce the release of openssh-2.1.0, the first stable
release of portable OpenSSH to incorporate support for the SSH2
protocol.

The SSH2 protocol offers a number of advantages over the SSH1 protocol
including standards compliance (SSH2 is on the IETF standards
track[1]), improved security and operation without RSA (which is
patented in some countries).

The SSH2 support in OpenSSH has been developed by Markus Friedl, with
support from the OpenBSD team.

This is also the first version of the portable version of OpenSSH
to offer built-in entropy collection. This removes the requirement
for EGD on systems that lack a /dev/random driver. As a result,
OpenSSH-2.1.0 now requires a recent version of OpenSSL[2] to compile
(version 0.9.5 or later).

NB. The portable version of OpenSSH is currently in the
process of merging its webpages with the official OpenBSD
project. Please use http://www.openssh.com/ from now
on. Distribution files are also available from the mirrors listed at
http://violet.ibs.com.au/openssh/files/MIRRORS.html

Please read http://www.openssh.com/report.html before reporting bugs.
Patches, bug reports, developer and user queries are welcome on the
mailing list (http://www.openssh.com/list.html).
Regards,
Damien Miller

[1] http://www.ietf.org/html.charters/secsh-charter.html
[2] http://www.openssl.org/
```

OpenSSH 1.2.3p1 (2000-03-24)

```
The Unix/Linux port of OpenSSH 1.2.3 was released yesterday and should
be available from a mirror near you. A mirror list is available from:

http://violet.ibs.com.au/openssh/files/MIRRORS.html

This release fixes the bugs reported since 1.2.2p1 and contains many
cleanups from the OpenBSD tree.

In particular, the OpenSSL detection problems have been resolved.

The layout has changed a little bit. The packages/ subdirectory has
been replaced with a contrib/ subdirectory which contains platform
specific code and other patches. Submissions are welcome.

Enjoy,
Damien Miller

20000317
 - Clarified --with-default-path option.
 - Added -blibpath handling for AIX to work around stupid runtime linking.
   Problem elucidated by gshapiro@SENDMAIL.ORG by way of Jim Knoble
   <jmknoble@pobox.com>
 - Checks for 64 bit int types. Problem report from Mats Fredholm
   <matsf@init.se>
 - OpenBSD CVS updates:
   - [atomicio.c auth-krb4.c bufaux.c channels.c compress.c fingerprint.c]
     [packet.h radix.c rsa.c scp.c ssh-agent.c ssh-keygen.c sshconnect.c]
     [sshd.c]
     pedantic: signed vs. unsigned, void*-arithm, etc
   - [ssh.1 sshd.8]
     Various cleanups and standardizations.
 - Runtime error fix for HPUX from Otmar Stahl
   <O.Stahl@lsw.uni-heidelberg.de>

20000316
 - Fixed configure not passing LDFLAGS to Solaris. Report from David G.
   Hesprich <dghespri@sprintparanet.com>
 - Propogate LD through to Makefile
 - Doc cleanups
 - Added blurb about "scp: command not found" errors to UPGRADING

20000315
 - Fix broken CFLAGS handling during search for OpenSSL. Fixes va_list
   problems with gcc/Solaris.
 - Don't free argument to putenv() after use (in setenv() replacement).
   Report from Seigo Tanimura <tanimura@r.dl.itc.u-tokyo.ac.jp>
 - Created contrib/ subdirectory. Included helpers from Phil Hands'
   Debian package, README file and chroot patch from Ricardo Cerqueira
   <rmcc@clix.pt>
 - Moved gnome-ssh-askpass.c to contrib directory and removed config
   option.
 - Slight cleanup to doc files
```

```
 - Configure fix from Bratislav ILICH <bilic@zepter.ru>

20000314
 - Include macro for IN6_IS_ADDR_V4MAPPED. Report from
   peter@frontierflying.com
 - Include /usr/local/include and /usr/local/lib for systems that don't
   do it themselves
 - -R/usr/local/lib for Solaris
 - Fix RSAref detection
 - Fix IN6_IS_ADDR_V4MAPPED macro

20000311
 - Detect RSAref
 - OpenBSD CVS change
   [sshd.c]
    - disallow guessing of root password
 - More configure fixes
 - IPv6 workarounds from Hideaki YOSHIFUJI <yoshfuji@ecei.tohoku.ac.jp>

20000309
 - OpenBSD CVS updates to v1.2.3
        [ssh.h atomicio.c]
         - int atomicio -> ssize_t (for alpha). ok deraadt@
        [auth-rsa.c]
         - delay MD5 computation until client sends response, free() early, cleanup.
        [cipher.c]
         - void* -> unsigned char*, ok niels@
        [hostfile.c]
         - remove unused variable 'len'. fix comments.
         - remove unused variable
        [log-client.c log-server.c]
         - rename a cpp symbol, to avoid param.h collision
        [packet.c]
         - missing xfree()
         - getsockname() requires initialized tolen; andy@guildsoftware.com
         - use getpeername() in packet_connection_is_on_socket(), fixes sshd -i;
        from Holger.Trapp@Informatik.TU-Chemnitz.DE
        [pty.c pty.h]
         - register cleanup for pty earlier. move code for pty-owner handling to
        pty.c ok provos@, dugsong@
        [readconf.c]
         - turn off x11-fwd for the client, too.
        [rsa.c]
         - PKCS#1 padding
        [scp.c]
         - allow '.' in usernames; from jedgar@fxp.org
        [servconf.c]
         - typo: ignore_user_known_hosts int->flag; naddy@mips.rhein-neckar.de
         - sync with sshd_config
        [ssh-keygen.c]
         - enable ssh-keygen -l -f ~/.ssh/known_hosts, ok deraadt@
        [ssh.1]
         - Change invalid 'CHAT' loglevel to 'VERBOSE'
        [ssh.c]
         - suppress AAAA query host when '-4' is used; from shin@nd.net.fujitsu.co.jp
         - turn off x11-fwd for the client, too.
        [sshconnect.c]
         - missing xfree()
         - retry rresvport_af(), too. from sumikawa@ebina.hitachi.co.jp.
         - read error vs. "Connection closed by remote host"
        [sshd.8]
         - ie. -> i.e.,
         - do not link to a commercial page..
         - sync with sshd_config
        [sshd.c]
         - no need for poll.h; from bright@wintelcom.net
         - log with level log() not fatal() if peer behaves badly.
         - don't panic if client behaves strange. ok deraadt@
         - make no-port-forwarding for RSA keys deny both -L and -R style fwding
         - delay close() of pty until the pty has been chowned back to root
         - oops, fix comment, too.
         - missing xfree()
         - move XAUTHORITY to subdir. ok dugsong@. fixes debian bug #57907, too.
        (http://cgi.debian.org/cgi-bin/bugreport.cgi?archive=no&bug=57907)       - register cleanup for pty earlier. move code for pty-owner
      pty.c ok provos@, dugsong@
         - create x11 cookie file
         - fix pr 1113, fclose() -> pclose(), todo: remote popen()
         - version 1.2.3
 - Cleaned up
 - Removed warning workaround for Linux and devpts filesystems (no longer
   required after OpenBSD updates)

20000308
 - Configure fix from Hiroshi Takekawa <takekawa@sr3.t.u-tokyo.ac.jp>
```

OpenSSH 1.2.2p1 (2000-03-05)

```
It gives me no little pleasure to announce the first stable release
of the Unix port of OpenSSH.

It is available in tar.gz and RPM format from one of the mirrors
listed at:

http://violet.ibs.com.au/openssh/files/MIRRORS.html
```

```
This release fixes all known issues and is known to compile and
function on (at least) recent releases on Linux, Solaris, HPUX and SCO
Unixware.

Please review the ChangeLog[1] for details on what has changed since
the last release.

I am holding off on a wider announcement until the mirrors have
updated.

Thanks to everyone who assisted with testing, bug reports, success
stories and most of all, patches :) Special thanks to the OpenBSD
developers for giving us OpenSSH to begin with.

Regards,
Damien Miller

[1] http://violet.ibs.com.au/openssh/files/ChangeLog
```

$OpenBSD: releasenotes.html,v 1.85 2026/04/15 21:06:39 tj Exp $