



[RELEASE](#)

## libssh 0.12.0 and 0.11.4 security releases

by [Jakub Jelen](#) | Published [February 10, 2026](#)

We are thrilled to announce a new libssh 0.12.0 release bringing several new features including Post-Quantum Crypto Key exchange mechanisms, support for FIDO2/U2F based keys, sshsig signatures (for example for git commits) and GSSAPI Key exchange.

Additionally, it fixes several security issues that were backported to libssh 0.11.4, our stable distribution.

A big thank you to all the contributors who made this release possible!

For those new to libssh, we recommend checking out our getting started [tutorial](#). If you have any questions, feel free to join our [mailing list](#) or visit our [Matrix channel](#).

You can download libssh-0.12.0 [here](#) and libssh-0.11.4 in [here](#).

## ChangeLog for libssh 0.12.0:

### Security:

- [CVE-2025-14821](#): libssh loads configuration files from the C:\etc directory on Windows
- [CVE-2026-0964](#): SCP Protocol Path Traversal in ssh\_scp\_pull\_request()
- [CVE-2026-0965](#): Possible Denial of Service when parsing unexpected configuration files
- [CVE-2026-0966](#): Buffer underflow in ssh\_get\_hexa() on invalid input
- [CVE-2026-0967](#): Specially crafted patterns could cause DoS
- [CVE-2026-0968](#): OOB Read in sftp\_parse\_longname()
- [libssh-2026-sftp-extensions](#): Read buffer overrun when handling SFTP extensions

### Deprecations and removals:

- Bumped minimal RSA key size to 1024 bits

### New functionality:

- Add support for hybrid key exchange mechanisms using Quantum Resistant cryptography for all backends. These are now preferred:
  - sntrup761x25519-sha512, sntrup761x25519-sha512@openssh.com
  - mlkem768nistp256-sha256
  - mlkem768x25519-sha256
  - mlkem1024nistp384-sha384 (only OpenSSL 3.5+ and libgcrypt)
- New cmake option WITH\_HERMETIC\_USR
- Added support for Ed25519 keys through PKCS#11
- Support for host-bound public key authentication (publickey-hostbound-v00@openssh.com)
- Use curve25519 implementation from mbedTLS and libgcrypt
- New functions for signing arbitrary data (commits) with SSH keys
  - sshsig\_sign()
  - sshsig\_verify()
- Support for FIDO/U2F keys (internal implementation using libfido2)
  - Compatible with OpenSSH: should work out of the box
  - Extensible with callbacks
- Add support for GSSAPI Key Exchange (RFC 4462, RFC 8732)
- Add support for new configuration options (client and server):
  - RequiredRsaSize
  - AddressFamily (client)
  - GSSAPIKeyExchange
  - GSSAPIKexAlgorithms
- New option to get list of configured identities (SSH\_OPTIONS\_NEXT\_IDENTITY)

- More OpenSSH compatible percent expansion characters
- Add new server auth\_kbdint\_function() callback
- New PKI Context structure for key operations
- Stability and compatibility improvements of ProxyJump

## SFTP

- Prevent failures when SFTP status message does not contain error message
- Fix possible timeouts while waiting for SFTP messages
- Support for users-groups-by-id@openssh.com extension in client
- Support for SSH\_FXF\_TRUNC in server

## ChangeLog for libssh 0.11.4

### Security:

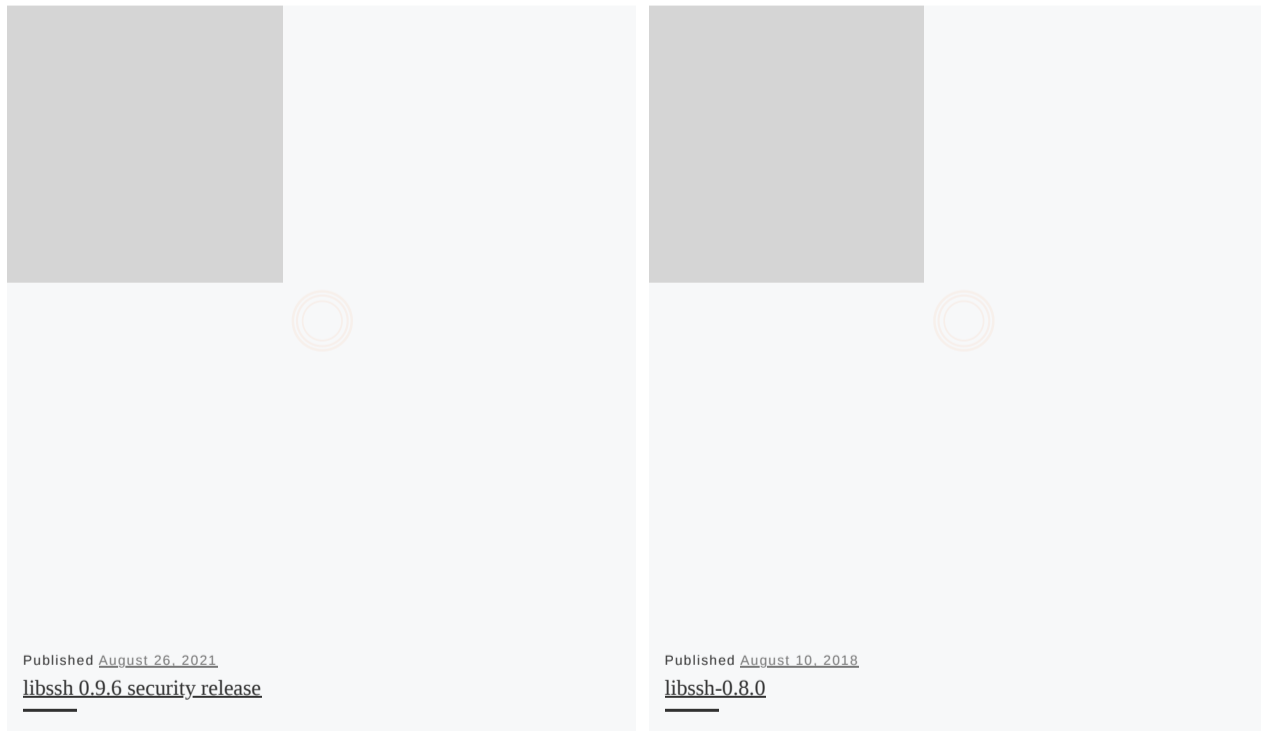
- [CVE-2025-14821](#): libssh loads configuration files from the C:\etc directory on Windows
- [CVE-2026-0964](#): SCP Protocol Path Traversal in ssh\_scp\_pull\_request()
- [CVE-2026-0965](#): Possible Denial of Service when parsing unexpected configuration files
- [CVE-2026-0966](#): Buffer underflow in ssh\_get\_hexa() on invalid input
- [CVE-2026-0967](#): Specially crafted patterns could cause DoS
- [CVE-2026-0968](#): OOB Read in sftp\_parse\_longname()
- [libssh-2026-sftp-extensions](#): Read buffer overrun when handling SFTP extensions

### Other fixes:

- Stability and compatibility improvements of ProxyJump

[cve](#) [Release](#)

### YOU MAY ALSO LIKE



Search ...



TAG CLOUD

- AES-GCM
- bugfix
- cve
- ed25519
- ETM
- FIPS
- gsoc
- gssapi
- gssproxy
- KDF
- Release
- security
- sftp
- sha2

[< WRAPPING UP GSOC 2025](#)



Projects using libssh ...



KDE uses libssh to implement the sftp module to allow secure file transfers between different computers.

GitHub



GitHub uses libssh in production to power its git SSH infrastructure, serving millions of requests daily.

Cockpit



Cockpit uses libssh for SSH public key authentication and to administer multiple machine using one bastion host.

© 2026 libssh - All rights reserved - The libssh logo has been created by [Robert Lihm](#)

Designed with [Customizr Pro](#)