NAME

scp — OpenSSH secure file copy

SYNOPSIS

scp [-346ABCOpqRrsTv] [-c *cipher*] [-D *sftp_server_path*] [-F *ssh_config*] [-i *identity_file*] [-J *destination*] [-l *limit*] [-o *ssh_option*] [-P *port*] [-S *program*] [-X *sftp_option*] *source ... target*

DESCRIPTION

scp copies files between hosts on a network.

scp uses the SFTP protocol over an *ssh*(1) connection for data transfer, and uses the same authentication and provides the same security as a login session.

scp will ask for passwords or passphrases if they are needed for authentication.

The *source* and *target* may be specified as a local pathname, a remote host with optional path in the form [user@]host:[path], or a URI in the form scp://[user@]host[:port][/path]. Local file names can be made explicit using absolute or relative pathnames to avoid **scp** treating file names containing ':' as host specifiers.

When copying between two remote hosts, if the URI format is used, a *port* cannot be specified on the *target* if the -R option is used.

The options are as follows:

-3 Copies between two remote hosts are transferred through the local host. This mode is the default, but see also the -R option for copying data directly between two remote hosts. Note that when using the legacy SCP protocol (via the -O flag), this option selects batch mode for the second host as **scp** cannot ask for passwords or passphrases for both hosts.

-4 Forces **scp** to use IPv4 addresses only.

-6 Forces **scp** to use IPv6 addresses only.

-A Allows forwarding of *ssh-agent*(1) to the remote system. The default is not to forward an authentication agent.

-B Selects batch mode (prevents asking for passwords or passphrases).

-C Compression enable. Passes the -C flag to *ssh*(1) to enable compression.

-c *cipher*
Selects the cipher to use for encrypting the data transfer. This option is directly passed to *ssh*(1).

-D *sftp_server_path*
Connect directly to a local SFTP server program rather than a remote one via *ssh*(1). This option may be useful in debugging the client and server.

-F *ssh_config*
Specifies an alternative per-user configuration file for **ssh**. This option is directly passed to *ssh*(1).

-i *identity_file*
Selects the file from which the identity (private key) for public key authentication is read. This option is directly passed to *ssh*(1).

-J *destination*
Connect to the target host by first making an **scp** connection to the jump host described by *destination* and then establishing a TCP forwarding to the ultimate destination from there. Multiple jump hops may be specified separated by comma characters. This is a shortcut to specify a ProxyJump configuration directive. This option is directly passed to *ssh*(1).

`-l` *`limit`*
Limits the used bandwidth, specified in Kbit/s.

`-O` Use the legacy SCP protocol for file transfers instead of the SFTP protocol. Forcing the use of the SCP protocol may be necessary for servers that do not implement SFTP, for backwards-compatibility for particular filename wildcard patterns and for expanding paths with a '~' prefix for older SFTP servers.

`-o` *`ssh_option`*
Can be used to pass options to **`ssh`** in the format used in *ssh_config*(5). This is useful for specifying options for which there is no separate **`scp`** command-line flag. For full details of the options listed below, and their possible values, see *ssh_config*(5).

AddKeysToAgent
AddressFamily
BatchMode
BindAddress
BindInterface
CASignatureAlgorithms
CanonicalDomains
CanonicalizeFallbackLocal
CanonicalizeHostname
CanonicalizeMaxDots
CanonicalizePermittedCNAMEs
CertificateFile
ChannelTimeout
CheckHostIP
Ciphers
ClearAllForwardings
Compression
ConnectTimeout
ConnectionAttempts
ControlMaster
ControlPath
ControlPersist
DynamicForward
EnableEscapeCommandline
EnableSSHKeysign
EscapeChar
ExitOnForwardFailure
FingerprintHash
ForkAfterAuthentication
ForwardAgent
ForwardX11
ForwardX11Timeout
ForwardX11Trusted
GSSAPIAuthentication
GSSAPIDelegateCredentials
GatewayPorts
GlobalKnownHostsFile
HashKnownHosts
Host
HostKeyAlgorithms

HostKeyAlias
HostbasedAcceptedAlgorithms
HostbasedAuthentication
Hostname
IPQoS
IdentitiesOnly
IdentityAgent
IdentityFile
IgnoreUnknown
Include
KbdInteractiveAuthentication
KbdInteractiveDevices
KexAlgorithms
KnownHostsCommand
LocalCommand
LocalForward
LogLevel
LogVerbose
MACs
NoHostAuthenticationForLocalhost
NumberOfPasswordPrompts
ObscureKeystrokeTiming
PKCS11Provider
PasswordAuthentication
PermitLocalCommand
PermitRemoteOpen
Port
PreferredAuthentications
ProxyCommand
ProxyJump
ProxyUseFdpass
PubkeyAcceptedAlgorithms
PubkeyAuthentication
RekeyLimit
RemoteCommand
RemoteForward
RequestTTY
RequiredRSASize
RevokedHostKeys
SecurityKeyProvider
SendEnv
ServerAliveCountMax
ServerAliveInterval
SessionType
SetEnv
StdinNull
StreamLocalBindMask
StreamLocalBindUnlink
StrictHostKeyChecking
SyslogFacility
TCPKeepAlive
Tag

Tunnel
TunnelDevice
UpdateHostKeys
User
UserKnownHostsFile
VerifyHostKeyDNS
VisualHostKey
XAuthLocation

`-P` *port*
: Specifies the port to connect to on the remote host. Note that this option is written with a capital 'P', because `-p` is already reserved for preserving the times and mode bits of the file.

`-p`
: Preserves modification times, access times, and file mode bits from the source file.

`-q`
: Quiet mode: disables the progress meter as well as warning and diagnostic messages from *ssh*(1).

`-R`
: Copies between two remote hosts are transferred through the local host by default. This option instead copies between two remote hosts by connecting to the origin host and executing **scp** there. This requires that **scp** running on the origin host can authenticate to the destination host without requiring a password.

`-r`
: Recursively copy entire directories. Note that **scp** follows symbolic links encountered in the tree traversal.

`-S` *program*
: Name of *program* to use for the encrypted connection. The program must understand *ssh*(1) options.

`-T`
: Disable strict filename checking. By default when copying files from a remote host to a local directory **scp** checks that the received filenames match those requested on the command-line to prevent the remote end from sending unexpected or unwanted files. Because of differences in how various operating systems and shells interpret filename wildcards, these checks may cause wanted files to be rejected. This option disables these checks at the expense of fully trusting that the server will not send unexpected filenames.

`-v`
: Verbose mode. Causes **scp** and *ssh*(1) to print debugging messages about their progress. This is helpful in debugging connection, authentication, and configuration problems.

`-X` *sftp_option*
: Specify an option that controls aspects of SFTP protocol behaviour. The valid options are:

  `nrequests=`*value*
  : Controls how many concurrent SFTP read or write requests may be in progress at any point in time during a download or upload. By default 64 requests may be active concurrently.

  `buffer=`*value*
  : Controls the maximum buffer size for a single SFTP read/write operation used during download or upload. By default a 32KB buffer is used.

EXIT STATUS

The **scp** utility exits 0 on success, and >0 if an error occurs.

SEE ALSO

sftp(1), *ssh*(1), *ssh-add*(1), *ssh-agent*(1), *ssh-keygen*(1), *ssh_config*(5), *sftp-server*(8), *sshd*(8)

HISTORY

scp is based on the rcp program in BSD source code from the Regents of the University of California.

Since OpenSSH 9.0, **scp** has used the SFTP protocol for transfers by default.

AUTHORS

Timo Rinne <*tri@iki.fi*>
Tatu Ylonen <*ylo@cs.hut.fi*>

CAVEATS

The legacy SCP protocol (selected by the `-O` flag) requires execution of the remote user's shell to perform *glob*(3) pattern matching. This requires careful quoting of any characters that have special meaning to the remote shell, such as quote characters.