NAME

plink – PuTTY link, command line network connection tool

SYNOPSIS

plink [*options*] [*user@*]*host* [*command*]

DESCRIPTION

plink is a network connection tool supporting several protocols.

OPTIONS

The command-line options supported by **plink** are:

-V Show version information and exit.

-pgpfp Display the fingerprints of the PuTTY PGP Master Keys and exit, to aid in verifying new files released by the PuTTY team.

-v Show verbose messages.

-load *session*
Load settings from saved session.

-ssh Force use of SSH protocol (default).

-telnet Force use of Telnet protocol.

-rlogin Force use of rlogin protocol.

-raw Force raw mode.

-serial Force serial mode.

–proxycmd *command*
Instead of making a TCP connection, use *command* as a proxy; network traffic will be redirected to the standard input and output of *command*. *command* must be a single word, so is likely to need quoting by the shell.

The special strings **%host** and **%port** in *command* will be replaced by the hostname and port number you want to connect to; to get a literal **%** sign, enter **%%**.

Backslash escapes are also supported, such as sequences like **\n** being replaced by a literal newline; to get a literal backslash, enter \\. (Further escaping may be required by the shell.)

(See the main PuTTY manual for full details of the supported **%**- and backslash-delimited tokens, although most of them are probably not very useful in this context.)

-P *port* Connect to port *port*.

-l *user* Set remote username to *user*.

-m *path*
Read remote command(s) from local file *path*.

-batch Disable interactive prompts.

-sanitise-stderr

-sanitise-stdout

-no-sanitise-stderr

-no-sanitise-stdout
By default, Plink can choose to filter control characters if that seems appropriate, to prevent remote processes sending confusing escape sequences. These options override Plink's default behaviour to enable or disabling such filtering on the standard error and standard output channels.

-pw *password*

Set remote password to *password*. *CAUTION:* this will likely make the password visible to other users of the local machine (via commands such as '**w**').

–L [*srcaddr***:]***srcport***:***desthost***:***destport*

Set up a local port forwarding: listen on *srcport* (or *srcaddr*:*srcport* if specified), and forward any connections over the SSH connection to the destination address *desthost*:*destport*. Only works in SSH.

–R [*srcaddr***:]***srcport***:***desthost***:***destport*

Set up a remote port forwarding: ask the SSH server to listen on *srcport* (or *srcaddr*:*srcport* if specified), and to forward any connections back over the SSH connection where the client will pass them on to the destination address *desthost*:*destport*. Only works in SSH.

–D [*srcaddr*:]*srcport*

Set up dynamic port forwarding. The client listens on *srcport* (or *srcaddr*:*srcport* if specified), and implements a SOCKS server. So you can point SOCKS-aware applications at this port and they will automatically use the SSH connection to tunnel all their connections. Only works in SSH.

-X Enable X11 forwarding.

-x Disable X11 forwarding (default).

-A Enable agent forwarding.

-a Disable agent forwarding (default).

-t Enable pty allocation (default if a command is NOT specified).

-T Disable pty allocation (default if a command is specified).

-1 Force use of SSH protocol version 1.

-2 Force use of SSH protocol version 2.

-4, **-6** Force use of IPv4 or IPv6 for network connections.

-C Enable SSH compression.

-i *keyfile*

Private key file for user authentication. For SSH-2 keys, this key file must be in PuTTY's PPK format, not OpenSSH's format or anyone else's.

If you are using an authentication agent, you can also specify a *public* key here (in RFC 4716 or OpenSSH format), to identify which of the agent's keys to use.

–noagent

Don't try to use an authentication agent for local authentication. (This doesn't affect agent forwarding.)

–agent Allow use of an authentication agent. (This option is only necessary to override a setting in a saved session.)

–noshare

Don't test and try to share an existing connection, always make a new connection.

–share Test and try to share an existing connection.

–hostkey *key*

Specify an acceptable host public key. This option may be specified multiple times; each key can be either a fingerprint (**99:aa:bb:...**) or a base64-encoded blob in OpenSSH's one-line format.

Specifying this option overrides automated host key management; *only* the key(s) specified on the command-line will be accepted (unless a saved session also overrides host keys, in which case those will be added to), and the host key cache will not be written.

-s Remote command is SSH subsystem (SSH-2 only).

-N Don't start a remote command or shell at all (SSH-2 only).

–nc *host*:*port*

Make a remote network connection from the server instead of starting a shell or command.

–sercfg *configuration-string*

Specify the configuration parameters for the serial port, in **-serial** mode. *configuration-string* should be a comma-separated list of configuration parameters as follows:

- Any single digit from 5 to 9 sets the number of data bits.
- '**1**', '**1.5**' or '**2**' sets the number of stop bits.
- Any other numeric string is interpreted as a baud rate.
- A single lower-case letter specifies the parity: '**n**' for none, '**o**' for odd, '**e**' for even, '**m**' for mark and '**s**' for space.
- A single upper-case letter specifies the flow control: '**N**' for none, '**X**' for XON/XOFF, '**R**' for RTS/CTS and '**D**' for DSR/DTR.

–sshlog *logfile*

–sshrawlog *logfile*

For SSH connections, these options make **plink** log protocol details to a file. (Some of these may be sensitive, although by default an effort is made to suppress obvious passwords.)

–sshlog logs decoded SSH packets and other events (those that **–v** would print). **–sshrawlog** additionally logs the raw encrypted packet data.

–shareexists

Instead of making a new connection, test for the presence of an existing connection that can be shared. The desired session can be specified in any of the usual ways.

Returns immediately with a zero exit status if a suitable 'upstream' exists, nonzero otherwise.

MORE INFORMATION

For more information on plink, it's probably best to go and look at the manual on the PuTTY web page:

https://www.chiark.greenend.org.uk/˜sgtatham/putty/

BUGS

This man page isn't terribly complete. See the above web link for better documentation.