Previous | Contents | Index | Next

Chapter 7: Using the command-line connection tool Plink

Plink (PuTTY Link) is a command-line connection tool similar to UNIX `ssh`. It is mostly used for automated operations, such as making CVS access a repository on a remote server.

Plink is probably not what you want if you want to run an interactive session in a console window.

7.1 Starting Plink

Plink is a command line application. This means that you cannot just double-click on its icon to run it and instead you have to bring up a console window. In Windows 95, 98, and ME, this is called an 'MS-DOS Prompt', and in Windows NT, 2000, and XP, it is called a 'Command Prompt'. It should be available from the Programs section of your Start Menu.

In order to use Plink, the file `plink.exe` will need either to be on your `PATH` or in your current directory. To add the directory containing Plink to your `PATH` environment variable, type into the console window:

```
set PATH=C:\path\to\putty\directory;%PATH%
```

This will only work for the lifetime of that particular console window. To set your `PATH` more permanently on Windows NT, 2000, and XP, use the Environment tab of the System Control Panel. On Windows 95, 98, and ME, you will need to edit your `AUTOEXEC.BAT` to include a `set` command like the one above.

7.2 Using Plink

This section describes the basics of how to use Plink for interactive logins and for automated processes.

Once you've got a console window to type into, you can just type `plink` on its own to bring up a usage message. This tells you the version of Plink you're using, and gives you a brief summary of how to use Plink:

```
Z:\sysosd>plink
PuTTY Link: command-line connection utility
Release 0.58
Usage: plink [options] [user@]host [command]
       ("host" can also be a PuTTY saved session name)
Options:
  -V        print version information and exit
  -pgpfp    print PGP key fingerprints and exit
  -v        show verbose messages
  -load sessname  Load settings from saved session
  -ssh -telnet -rlogin -raw
            force use of a particular protocol
  -P port   connect to specified port
  -l user   connect with specified username
  -batch    disable all interactive prompts
The following options only apply to SSH connections:
  -pw passw login with specified password
  -D [listen-IP:]listen-port
            Dynamic SOCKS-based port forwarding
  -L [listen-IP:]listen-port:host:port
            Forward local port to remote address
  -R [listen-IP:]listen-port:host:port
            Forward remote port to local address
  -X -x     enable / disable X11 forwarding
  -A -a     enable / disable agent forwarding
  -t -T     enable / disable pty allocation
  -1 -2     force use of particular protocol version
  -4 -6     force use of IPv4 or IPv6
  -C        enable compression
  -i key    private key file for authentication
  -m file   read remote command(s) from file
  -s        remote command is an SSH subsystem (SSH-2 only)
  -N        don't start a shell/command (SSH-2 only)
```

Once this works, you are ready to use Plink.

7.2.1 Using Plink for interactive logins

To make a simple interactive connection to a remote server, just type `plink` and then the host name:

```
Z:\sysosd>plink login.example.com
```

```
Debian GNU/Linux 2.2 flunky.example.com
flunky login:
```

You should then be able to log in as normal and run a session. The output sent by the server will be written straight to your command prompt window, which will most likely not interpret terminal control codes in the way the server expects it to. So if you run any full-screen applications, for example, you can expect to see strange characters appearing in your window. Interactive connections like this are not the main point of Plink.

In order to connect with a different protocol, you can give the command line options `-ssh`, `-telnet`, `-rlogin` or `-raw`. To make an SSH connection, for example:

```
Z:\sysosd>plink -ssh login.example.com
login as:
```

If you have already set up a PuTTY saved session, then instead of supplying a host name, you can give the saved session name. This allows you to use public-key authentication, specify a user name, and use most of the other features of PuTTY:

```
Z:\sysosd>plink my-ssh-session
Sent username "fred"
Authenticating with public key "fred@winbox"
Last login: Thu Dec  6 19:25:33 2001 from :0.0
fred@flunky:~$
```

(You can also use the `-load` command-line option to load a saved session; see section 3.7.3.1. If you use `-load`, the saved session exists, and it specifies a hostname, you cannot also specify a `host` or `user@host` argument - it will be treated as part of the remote command.)

7.2.2 Using Plink for automated connections

More typically Plink is used with the SSH protocol, to enable you to talk directly to a program running on the server. To do this you have to ensure Plink is *using* the SSH protocol. You can do this in several ways:

- Use the `-ssh` option as described in section 7.2.1.
- Set up a PuTTY saved session that describes the server you are connecting to, and that also specifies the protocol as SSH.
- Set the Windows environment variable `PLINK_PROTOCOL` to the word `ssh`.

Usually Plink is not invoked directly by a user, but run automatically by another process. Therefore you typically do not want Plink to prompt you for a user name or a password.

Next, you are likely to need to avoid the various interactive prompts Plink can produce. You might be prompted to verify the host key of the server you're connecting to, to enter a user name, or to enter a password.

To avoid being prompted for the server host key when using Plink for an automated connection, you should first make a *manual* connection (using either of PuTTY or Plink) to the same server, verify the host key (see section 2.2 for more information), and select Yes to add the host key to the Registry. After that, Plink commands connecting to that server should not give a host key prompt unless the host key changes.

To avoid being prompted for a user name, you can:

- Use the `-l` option to specify a user name on the command line. For example, `plink login.example.com -l fred`.
- Set up a PuTTY saved session that describes the server you are connecting to, and that also specifies the username to log in as (see section 4.14.1).

To avoid being prompted for a password, you should almost certainly set up public-key authentication. (See chapter 8 for a general introduction to public-key authentication.) Again, you can do this in two ways:

- Set up a PuTTY saved session that describes the server you are connecting to, and that also specifies a private key file (see section 4.20.5). For this to work without prompting, your private key will need to have no passphrase.
- Store the private key in Pageant. See chapter 9 for further information.

Once you have done all this, you should be able to run a remote command on the SSH server machine and have it execute automatically with no prompting:

```
Z:\sysosd>plink login.example.com -l fred echo hello, world
hello, world

Z:\sysosd>
```

Or, if you have set up a saved session with all the connection details:

```
Z:\sysosd>plink mysession echo hello, world
hello, world

Z:\sysosd>
```

Then you can set up other programs to run this Plink command and talk to it as if it were a process on the server machine.

7.2.3 Plink command line options

Plink accepts all the general command line options supported by the PuTTY tools. See section 3.7.3 for a description of these options.

Plink also supports some of its own options. The following sections describe Plink's specific command-line options.

7.2.3.1 `-batch`: disable all interactive prompts

If you use the `-batch` option, Plink will never give an interactive prompt while establishing the connection. If the server's host key is invalid, for example (see section 2.2), then the connection will simply be abandoned instead of asking you what to do next.

This may help Plink's behaviour when it is used in automated scripts: using `-batch`, if something goes wrong at connection time, the batch job will fail rather than hang.

7.2.3.2 `-s`: remote command is SSH subsystem

If you specify the `-s` option, Plink passes the specified command as the name of an SSH 'subsystem' rather than an ordinary command line.

(This option is only meaningful with the SSH-2 protocol.)

7.3 Using Plink in batch files and scripts

Once you have set up Plink to be able to log in to a remote server without any interactive prompting (see section 7.2.2), you can use it for lots of scripting and batch purposes. For example, to start a backup on a remote machine, you might use a command like:

```
plink root@myserver /etc/backups/do-backup.sh
```

Or perhaps you want to fetch all system log lines relating to a particular web area:

```
plink mysession grep /~fred/ /var/log/httpd/access.log > fredlog
```

Any non-interactive command you could usefully run on the server command line, you can run in a batch file using Plink in this way.

7.4 Using Plink with CVS

To use Plink with CVS, you need to set the environment variable `CVS_RSH` to point to Plink:

```
set CVS_RSH=\path\to\plink.exe
```

You also need to arrange to be able to connect to a remote host without any interactive prompts, as described in section 7.2.2.

You should then be able to run CVS as follows:

```
cvs -d :ext:user@sessionname:/path/to/repository co module
```

If you specified a username in your saved session, you don't even need to specify the 'user' part of this, and you can just say:

```
cvs -d :ext:sessionname:/path/to/repository co module
```

7.5 Using Plink with WinCVS

Plink can also be used with WinCVS. Firstly, arrange for Plink to be able to connect to a remote host non-interactively, as described in section 7.2.2.

Then, in WinCVS, bring up the 'Preferences' dialogue box from the *Admin* menu, and switch to the 'Ports' tab. Tick the box there labelled 'Check for an alternate `rsh` name' and in the text entry field to the right enter the full path to `plink.exe`. Select 'OK' on the 'Preferences' dialogue box.

Next, select 'Command Line' from the WinCVS 'Admin' menu, and type a CVS command as in section 7.4, for example:

```
cvs -d :ext:user@hostname:/path/to/repository co module
```

or (if you're using a saved session):

```
cvs -d :ext:user@sessionname:/path/to/repository co module
```

Select the folder you want to check out to with the 'Change Folder' button, and click 'OK' to check out your module. Once you've got modules checked out, WinCVS will happily invoke plink from the GUI for CVS operations.

If you want to provide feedback on this manual or on the PuTTY tools themselves, see the Feedback page.

[PuTTY release 0.58]